Multiray: Multi-Finger Raycasting for Large Displays

Fabrice Matulic[1,2] **Daniel Vogel**[2]
[1]Preferred Networks Inc., Japan [2]University of Waterloo, Canada
{fabrice.matulic, dvogel}@uwaterloo.ca

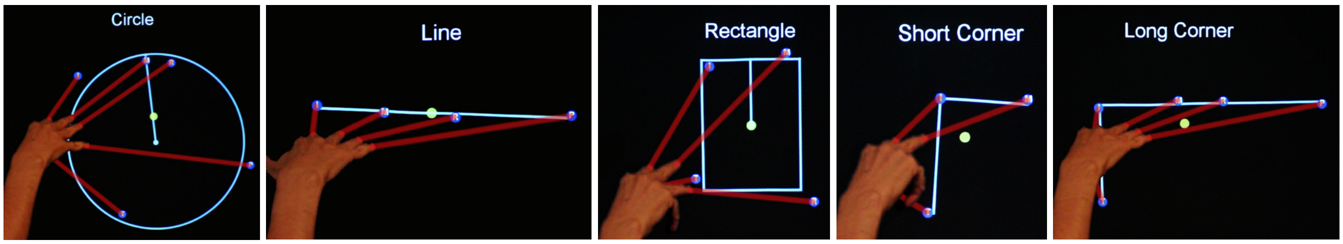


Figure 1. Shapes detected when aligning multi-finger raycast projections in formations (red projection lines for illustration only)

ABSTRACT
We explore and evaluate a multi-finger raycasting design space that we call "multiray". Each finger projects a ray on to the display, so the user is interacting from a distance using a form of direct input. Specifically, we propose techniques, where patterns of ray intersections created by hand postures form 2D geometric shapes to trigger actions and perform direct manipulations that go beyond single-point selections. Two formative studies examine characteristics of multi-finger raycasting for different projection methods, shapes, and tasks. Based on the results of those investigations, we demonstrate a number of dynamic UI controls and operations that utilise multiray points and shapes.

Author Keywords
mid-air gestures, freehand gestures, large displays, pointing

ACM Classification Keywords
H.5.2. User Interfaces-Interaction Techniques

INTRODUCTION
An intuitive method to select objects on a large display from a distance is simply to point. The target object is determined by computing the intersection of a ray emanating from the pointing device through the display plane. The ray can be produced physically using a laser pointer [4, 30], or with 3D tracking equipment to produce virtual rays. Devices like Wiimotes [28, 35], optically tracked wands [18, 20, 21, 31] and chopsticks [7] can also be used. Without a device, the most intuitive instrument is the index finger. Different projection methods and techniques that involve index finger raycasting have been proposed and evaluated [10, 33, 41].

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

ACM ISBN 978-1-4503-5620-6/18/04…$15.00
https://doi.org/10.1145/3173574.3173819

Except for a few techniques using two rays [4, 48], the focus of hand-based raycasting has been single ray pointing. When other fingers are involved, their position relative to the hand is used to change modes or trigger actions like finger "clicks" [10, 32, 37, 41]. Aside from index finger raycast pointing, no other mid-air hand posture projects rays onto the display for direct input. The dominant paradigm is to transform relative 3D motion, such as waving or grabbing, into indirect input for manipulations and command execution [25, 38, 42].

This dominance of single-finger direct input at a distance is in stark contrast to multitouch input. The ability to use more than one finger for direct touch input was recognised as a major improvement over single touch technology, and in research, interaction techniques using all fingers have been developed [12, 13, 29, 43] including several that leverage the formation of shapes [17, 29, 44, 46, 49]. Elicitation studies even show people would like to transfer multitouch gestures to distant interaction with large displays [10, 22, 45]. Therefore, we think multi-finger raycasting on large displays merits investigation.

In this paper, we explore and evaluate a multi-finger raycasting design space we call "multiray". The central idea is that each finger projects a ray on to the screen, so the user is interacting with a large display from a distance with direct input. Specifically, we propose techniques, where patterns of ray intersections created by hand postures form 2D geometric shapes to trigger actions and perform direct manipulations that go beyond single-point selections. We conduct two formative studies looking at the practicality of multiray for different projection methods, shapes and tasks. We contribute comparisons and analyses of various influencing factors, including fingertip versus knuckle-based projection, multi-finger single-point targeting, horizontal versus vertical shapes, posture registration performance, maintainability through motions and shape breaking as well as finger stability. Based on the results of those investigations, we demonstrate several examples of dynamic UI controls and operations that leverage multiray points and shapes.

RELATED WORK

Motivated by early work on raycasting for virtual environments [36] and using laser pointers [30], Vogel and Balakrishnan initiated an exploration of the design space of freehand pointing for large vertical displays [41]. In particular, they identified a desirability for switching from absolute raycasting to relative pointing for increased precision and performance. Other authors followed by proposing and evaluating other techniques for target acquisition using relative cursor control [6, 21]. While having non-absolute mappings to improve single-point selection makes sense, it is less obvious how those techniques translate to multi-finger raycasting, especially those using transfer functions that dynamically modify the CD gain [19, 26]. Presumably, such techniques might be cognitively disruptive as cursor speeds and mappings would be different for each finger at a given time.

A related issue is how to construct the projection ray. For absolute directional raycasting, two points are necessary to form the ray. While the second point is almost invariably the finger or device tip, several options have been tried for the origin. These include hand-, forearm-, head- and body-rooted origins, sometimes with offsets to compensate for parallax [2, 4, 5, 18, 27]. Many of those techniques are unlikely to be directly transferable to multi-finger raycast because a single origin chosen for all finger tips would distort the multiple rays. This mismatch was acknowledged by Banerjee et al. when designing their thumb+index raycast technique [4], as they chose to use an absolute projection for the index finger and a relative projection for the thumb (based on the distance and angle between the two fingers). This problem is likely compounded when more than two fingers are involved.

A popular type of non-raycast projection considers invisible tracking surfaces (or volumes) in front of the user. The boundaries of those surfaces or volumes are mapped to those of a large display, or virtual 3D world, to enable pointing [9, 33]. While less prone to jitter, such spatial mappings use fixed tracking areas, limiting the interaction range and increasing fatigue [9, 50]. Other techniques like Myopoint [16] and SmartCasting [34] mimic raycasting using IMUs, but are actually relative cursor input.

Finally, because raycasting is a continuous action that does not inherently include an explicit state-changing gesture like touch, specific interactions are required to perform clicks, clutches and other binary switches. A straightforward solution that is used in many applications and Kinect games is dwelling on the target. With just a single ray emanating from the index finger, however, it is also possible to use other fingers as mode-triggers. Two of the most popular gestures are thumb-on-index pinches and mid-air taps [10, 32, 37, 41]. If the cursor is mapped to the whole hand, clenching/grabbing and pushing a virtual button are also common choices [3, 35, 41]. Furthermore, when bimanual interaction is supported, mode-changes and commands can be activated by the non-dominant hand, which reduces displacements of the pointer controlled by the dominant hand [5, 31].

MULTIRAY: MULTI-FINGER RAYCASTING

Motivated by multitouch techniques, in particular chords [12, 13, 23, 43], we explore a "multiray" space of multi-finger raycasting using chord-inspired interactions in mid-air. As noted above, non-hand-rooted ray projections or CD-gain modulated relative input with multiple cursors are not well-suited for multiple rays. We adopt a configuration where each finger directly and independently emits an unmodified ray. This is in line with initial investigations of absolute raycasting using laser pointers. Our approach is motivated by imagining a hand with lasers attached to each finger, similar to laser gloves used for light shows and special effects [1]. Our first multiray device prototype was in fact a self-made "glove" consisting of laser pointers attached to the fingertips.

Single Point Targeting

While single-point targeting is not our main focus, we are interested in examining how multiray may affect pointing performance. In multitouch, Moscovich and Hughes demonstrated multiple fingers controlling a cursor brought increased movement fluidity [29]. We would like to investigate if using several fingers for distant cursor control might improve acquisition stability, which is known to be one of the main factors affecting targeting speed due to jitter [6].

Shape Forming

Multitouch chords are usually defined as mere finger combinations, not as the geometric shapes the connected contact points form. With multitouch, maintaining finger contact with the surface likely constrains the kind of shapes that can be comfortably achieved and occlusion is also an issue. Multiray does not have those restrictions, so associating specific spatial finger formations with easily identifiable geometric shapes seems promising. Furthermore, mapping functions to simple shapes, such as lines, circles, and rectangles, may be easier to memorise than less visually distinct chords.

Simple, recognisable geometric shapes can be formed by a minimum of two raycast points (short segment), but more commonly three or more. Following multitouch standards, two raycast points are used for rotation-scale-translation (RST) manipulations. We select five other shapes corresponding to clear identifiable geometric shapes that can be easily formed with the hand (at least without raycasting) without contorting fingers in unnatural positions (Figure 1):

- *Short Corner*: A right angle formed by the thumb, the index finger and the middle finger, where the right angle is at the index finger.
- *Long Corner*: Like the short corner, but with all five fingers. All fingers but the thumb need to be aligned.
- *Line*: A Line segment formed by four or five fingers.
- *Rectangle*: A rectangle with all but the little finger placed at the four corners of the shape.
- *Circle*: The five fingers are roughly placed on a circle

To form a shape, the user points the required number of fingers at the screen and attempts to align the projected points according to a shape recognition template. When points match the template within a certain tolerance, the corresponding function visualisation appears on the display. The matching requirements depend on the recognition algorithm. We use ad hoc features and metrics based on least-squares error to estimate the quality of fit of a shape and/or of the right angles of corners. Depending on desired flexibility, shape templates can have specific orientations, sizes, or aspect ratios. This theoretically enables a single shape to trigger multiple commands or modes. We investigate human ability to form shapes using multiray in the experiments that follow.

Projection Method

To our knowledge, no prior work has experimentally compared different hand-rooted projections. We consider two projection vectors both shooting off the fingertip, but originating from different anchor points (illustrated in Figure 2). The first projection method, from the distal interphalangeal joint (DIP) (or interphalangeal joint for the thumb), is perhaps the most intuitive as it mimics rays from lasers attached to the fingertips. It theoretically allows for maximum range and control, but likely at the cost of stability. The second is projecting from the metacarpophalangeal joint (MCP). We hypothesise this provides less control but is more stable, since the points that define the direction vector are further apart. We refer to these projection methods as DIP and MCP.

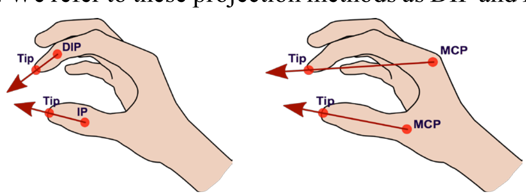


Figure 2. Different fingertip raycast projections: (left) from DIP/IP joint to tip; (right) from MCP joint to tip.

EXPERIMENT 1

The goal of this first study is to assess pointing and shape-forming performance with the DIP and MCP projections.

Participants

We recruited 8 right-handed participants from our university: average age 25.4 years (SD=4.5), 2 were females. Four participants had some experience with mid-air gestures using Leap Motion or Kinect devices. Remuneration was CAD 20.

Apparatus and Setup

Although bare hand tracking algorithms exist [40], small tracking errors are amplified by ray-based projection. To mitigate such errors, we use a 10-camera Vicon motion-capture system with retroreflective markers. Using double-sided tape, we attached nineteen 5mm hemispherical markers to the finger joints, and one 10 mm spherical marker to the back of the hand (see Figure 3a). These markers are sufficiently small and light to not cause impediments, and the feeling is like interacting with a bare hand. A hand model with labelled joints tracked using Vicon Nexus enables each marker (and finger) to be identified. The tool tailors the model to different hand morphologies using a "functional calibration". This procedure captures a short sequence of hand movements, then optimises the model's joint and marker positions for that person. Once calibrated, labelled marker data was streamed at 120FPS to custom C# and Java applications running on a Windows PC. Individual finger projections were identified with the finger initial (e.g. T for thumb, I for index finger) shown inside the projected dot as shown on Figure 6.

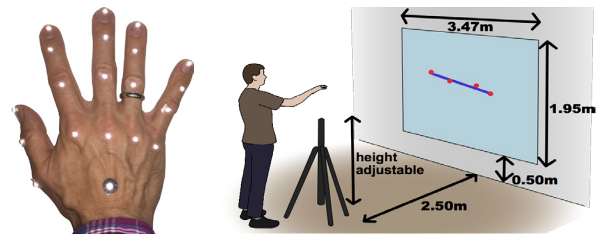


Figure 3. Apparatus: (a) marker placement on the hand; (b) locations of participant, registration tripod, and large display (cameras and projector not represented).

Digital content was projected on a white wall using a full HD short-throw projector. A tripod was placed 2.5m from the wall behind which participants were asked to stand. Other relevant distances and lengths are shown in Figure 3b.

System Noise and Filtering

To measure the effect of tracking noise on raycasting, we placed a self-made motion-capture glove with the same hand marker setup on a platform on the tripod. We recorded the displacements of points raycast from the index finger from the centroid in a 10 second window. For MCP projection, we obtained a mean displacement of 0.7mm (SD=0.5mm) and a max value of 3.6mm. For DIP projection, the respective measurements were 1.2mm (SD=0.7) and 5.4mm. Note that noise can increase, sometimes drastically, if the cameras are not able to reliably track the markers. This is in addition to the jitter resulting from shaky user hands and fingers. To compensate for noise and jitter, we applied independent 1€ filters [8] to each projected point with parameters 0.8 for cutoff and 0.002 for beta. This reduced noise to 0.2mm mean (SD=0.1) and 0.7mm max displacements for MCP and 0.4mm mean (SD=0.2) and 0.9mm max for DIP. The filters were active throughout the two studies.

Tasks

Based on the experiment goals, we designed the study around three separate tasks: single-point targeting; shape elicitation; and shape matching.

Single-point targeting

This task is a simplified form of classic distant pointing studies in which participants point at circular "dot" targets. To allow direct comparisons of a single dependent variable, we fix the accuracy requirement by setting the distance-to-target tolerance error to a radius of 2.7cm (15px) around the dot and measure the time taken to point at and dwell 1s on the dot within that tolerance threshold.

We tested targeting with three different techniques, each using a different number of finger rays: classic single index finger raycast (1F), two-finger raycast using the index finger and the thumb (2F), and five-finger raycast (5F). For multiple finger pointing, the centroid of the projected points was used as the selection cursor.

Nine target positions were used, one central and eight located at cardinal points, 25% and 75% of the screen width and height. The positions were presented in blocks using a fixed order: the central target, the rightmost target, then the remaining targets in clockwise order. For each projection method and technique, the participant trained on one block (with the option to repeat), then they completed two measured blocks. In summary: 9 targets × 2 blocks × 3 techniques × 2 projection methods × 8 participants = 864 data points.

Shape Elicitation

The purpose of this task was to elicit size, orientation and aspect ratios of given shapes from participants, when not explicitly prompted with a matching target by the system. To achieve that, the experimenter orally described the shape with the constraints to obey and participants formed that shape as they desired on the screen. In order not to bias participants, no visual feedback other than the projected points was shown on the screen, that is, no shapes or lines connecting points were displayed. The shapes to form were those described above. Participants signalled the experimenter when they were satisfied with the produced shape so that point positions could be logged. This process was repeated for each shape and for each projection method.

Shape Matching

The shape-matching task followed a pattern similar to the single-point targeting task. A series of shape targets were projected on the screen, which participants had to match by overlapping the shape produced by the raycast points. For corner and extremity points, the cast dots had to be on, or close to the corresponding point on the target. For all the circle points and the inner segment points of *Line* and *Long Corner* shapes, the dots could be placed anywhere on the contour or segment. Because multiple points needed to be matched simultaneously for the shape to be validated, the distance tolerance threshold was increased to 10.9cm (60px).

In terms of size, the target shapes were chosen so that their longer edge did not exceed 90.5cm and their shorter edge was a minimum of 22.6cm. Thus, non-line shapes were contained in a bounding box with an area roughly equal to 205cm². For *Corner* and *Rectangle* targets, the aspect ratio recorded in the shape elicitation phase was used. All shapes were axis-aligned and we created vertical and horizontal versions (denoted by V and H) of *Line* and the two *Corner* shapes.

In total, participants had to match eight types of shapes: *Line H* and *V*, *Short Corner H* and *V*, *Long Corner H* and *V*, *Circle* and *Rectangle*. Due to the number of shapes to test, we limited the target locations to five: one central target and four targets at the corners with the centroid of the target at 25% and 75% of the screen width and height again. Thus, the total number of matching trials for this task was 5 trials × 2 blocks × 8 shapes × 2 projection methods × 8 participants = 1352. As with the single-point targeting task, participants also trained with one block of trials for each shape and projection method before performing the main task.

Protocol and Design

Except for the shape elicitation task, all task trials required participants to start by gripping a 4cm diameter tripod column with the thumb resting on the top. We refer to this as the neutral position. Participants could adjust the height of the column so that this pose was comfortable for them. Trial sequences consisted in holding the neutral position, performing the trial, returning to the neutral position, performing the next trial and so on. The system automatically detected if a participant's hand was in the neutral position using marker positions recorded in an initial calibration step. Time measurements were started when the hand left the neutral position and stopped when targeting was achieved. Participants were given the opportunity to take breaks between each task set.

The study followed a within-subjects design in which each task was repeated for the two projection methods. Each task was completed with both projection methods before moving on to the next task. The order of the projection methods and techniques/shapes were counterbalanced between participants and between tasks.

Results

A session lasted for approximately two hours, where roughly 45 minutes was taken by preparation.

Single-point targeting

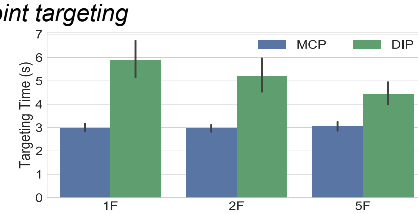


Figure 4. Mean targeting times for each technique and projection method (error bars represent standard deviation)

Figure 4 shows the mean targeting times after removing outliers (values more than three times the standard deviation away from the mean). An ANOVA on Technique × Projection Method with mean targeting times only yielded main effects for Projection Method: $F(1, 7) = 17.6$, $p = 0.004$. Due to the lower amount of noise, participants were able to target dots significantly faster using MCP projection. We had hoped to demonstrate that targeting with the centroid of multiple finger projections could help compensate for single-finger cursor instability, but the differences for Technique with DIP are not statistically significant: $F(2, 14) = 1.69$, $p = 0.22$. Other than the low number of participants, a possible reason why timings using multiple fingers are not lower is that all finger points had to be on the screen for the centroid to appear. For non-central targets, it sometimes took additional time for participants to cast all required points on the display.

Shape Elicitation

For reasons of space, we report the combined results of this task with the same task, which was repeated in Study 2, in the latter's section further below.

Shape Matching

Figure 5 shows the mean matching times of all shapes for the two projection methods. ANOVAs for our two independent variables with the mean matching time for each participant were conducted with Greenhouse-Geisser corrections when the sphericity condition was violated. We obtained main effects for both variables: FS(2.63, 18.43) = 4.3, p = 0.0011 and FPM(1, 7) = 26.86, p = 0.0013. Hence, we can conclude again that DIP led to significantly better performance. The results of pairwise comparisons with Bonferroni corrections revealed that *Short Corner H* and *Line H* were quicker to form compared to *Long Corner H* (p=0.004 and p=0.009 respectively). The main trend that we can observe is that *Long Corners* and *Rectangle* are the slowest and *Short Corners* the fastest shapes with good performances for *Line H* and *Circle* using MCP. Horizontal shapes were also formed faster.

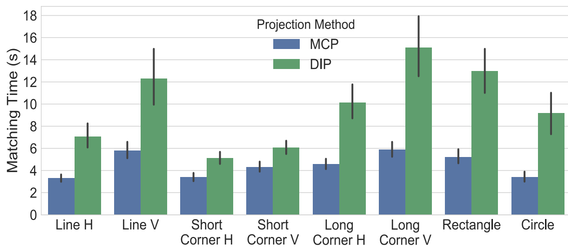


Figure 5. Mean matching times (including dwelling time of 1s) for each technique and projection method

Participant Feedback

In their subjective ratings of the projection methods, all participants but one either preferred MCP over DIP or indicated no preference. MCP was generally perceived as considerably more stable. For the first pointing task, even though the results did not show statistically significant differences, three participants stated that they felt using more fingers, especially five, brought more stability. For the shape-matching task, two participants commented that there was a mismatch between the physical hand posture and the projected shape on the screen. Initially, some participants primarily looked at the finger formation of their hand and expected the projected points to align accordingly on the screen. This is however not the most effective strategy. It is best to focus directly on the raycast points and adjust fingers individually to achieve the desired shape. This is similar to the mental model of indirect cursor control with a mouse, but with multiple cursors. When given that tip, participants performed better.

Discussion

The results show a quantitative and qualitative advantage for MCP projection, hence we adopt MCP as the projection method going forward. We now discuss a number of practical considerations about forming shapes with multiray.

Relaxed Shape-Forming

We imagine associations of detected shapes with particular UI tools for which location matters more than their exact size. For example, for a selection marquee or a zooming lens associated with a *Circle*, it is more important that the desired elements to select or view are included within the shape than the shape being of very specific dimensions. The same argument applies for a *Line* used as a ruler or an alignment tool. We therefore consider a set of requirements for shape-matching based on four constraints: appearance, location, orientation and minimum size. This means that lines are infinite, i.e. projected points need only be aligned on the target line, and only the location point needs to be targeted precisely for other shapes. This point is: the centre for *Rectangle* and *Circle* and the right-angle point for the *Corner* shapes.

Shape Maintainability during Motion

In some applications of multi-raycasting, it might be important not only to cast shapes at a particular location, but also to be able to comfortably move them to other areas without breaking the formation. Using the circle zoom lens example again, it should be possible to move the lens over a view to inspect elements without it disappearing because the fingers could not maintain the circle shape. Some degree of detection relaxation may be possible [47], but there are situations where detection must continue during movement.

Independent Finger Control

After a shape has been detected or registered, it can be desirable to have some controls available, for example, to validate or cancel the operation associated with that shape or to modify one of its parameters. In the single-raycast case, that can be done with gestures performed by non-pointing fingers. In multi-raycast, we have to be more careful, as several fingers participate in the formation of shapes. One solution is to define a subset of the shape points as anchor points that determine the position of the shape after it has been registered, while the other points are used as controls. In that case, the shape-preservation constraints are relaxed and can be limited, for example, to simply maintaining the number of fingers required by the shape on screen or within a specific zone surrounding the shape (see next subsection). Concretely, for a shape like *Rectangle*, this could mean that after registering the shape with all four corner points at right angles, one finger could be used as an independent control, since three points suffice to determine a rectangular shape (and even two if the rectangle is axis-aligned). Considering fingers naturally shake, we need to determine a sufficiently large safe zone that users need to explicitly exit to trigger the control.

There have been a number of studies that have assessed individual finger movement capabilities [14, 15, 24, 39]. The general consensus among those studies is that the middle and ring fingers are the least individuated and the index finger is the most independent. As underlined by Gracia-Ibáñez et al. [14], however, the mobility of finger joints depends on the posture of the adjacent ones. In our case, we have postures that considerably differ from those used in the experiments of prior work. Some of those postures even include fingers possibly hindering the movement of others, so we would like to conduct our own focused evaluation of finger stability.

Finger-Count Breaking

An issue related to the previous one is the feasibility for a raycast point not required for a shape to be kept far away from it. In other words, is it possible to keep the projected points of non-participating fingers distant from the points actually forming the shape? There are two reasons why that might be useful: 1) If the sensing technology cannot associate projected points to their originating fingers, it is easier to detect shapes without distractors. 2) Non-participating finger points kept afar can have a "breaking" function when moved closer to a relaxed shape kept active by finger count. Specifically, by entering the zone of a formed shape, the additional finger point can "break" its finger count requirement. This event can be used to trigger a particular action and/or dismiss the shape. As an example of this interaction, we can imagine a simple two-finger RST operation, which could be validated by bringing in a third finger into the breaking zone.

Since we only consider single hands in this work, finger-count breaking gestures naturally only concern shapes that use fewer than five fingers. Determining how far people can keep non-participating fingers away from posture shapes helps us to define such breaking zones.

FORMATIVE STUDY 2

In this second formative study, we move to conditions that more closely reflect how we envisage applications of multi-finger raycasting. We would like to investigate shape-matching again, but with relaxed conditions as well as maintainability during motion and individual finger dexterity.

Participants

We recruited 22 right-handed participants from our university (7 females, 15 males), average age 25.8 years (SD=5.6), including the eight participants from Study 1. Six participants had some experience with mid-air gestures, one person had participated in a kinesiology study involving body and hand tracking, and two participants were regular musicians (a clarinettist and a pianist). Despite lengthy training, one participant was unable to form the shapes, hence we collected data from 21 people only. Compensation was CAD 20.

Apparatus and Setup

The apparatus and setup was the same as in Experiment 1.

Tasks

The three tasks follow directly from the dynamic aspects of shape formation identified in the discussion of Experiment 1.

Shape Elicitation

To obtain a larger number of user-formed shapes without any prior bias (except with participants of Study 1), we conducted the shape elicitation experiment in Study 2 as well. The task design was the same, but with only one projection method.

Shape Matching and Moving

In this version of the shape-matching task, participants were first presented with shape targets at four cardinal points (left, right, top, bottom) on the screen (i.e. there were no central targets), which they had to match based on the relaxed conditions described above. Circles and rectangles now had an orientation constraint to be met (Figure 6). After matching the shape, a second target was shown on the other side of the screen to which the shape had to be moved while maintaining the posture. If the shape formation was broken during the motion phase, the fitted shape turned purple and a miss was recorded. When the translation was completed with the second target being matched, a third target shape appeared at the original location of the first target. Thus, a trial consisted of: 1) Initial matching of first target. 2) Moving the shape to the other side of the screen and matching the second target. 3) Moving the shape back to the original target and matching it. Motions for each trial were either horizontal or vertical back and forth translations.

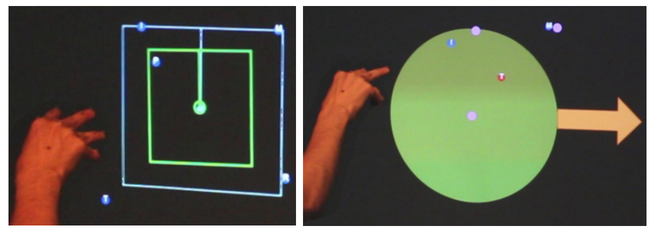


Figure 6. Left: Matching a rectangle target with orientation constraint (shown by vertical line). The target shape turns green to show that the matching requirements are met. Right: Finger-moving trial for Short Corner V in which the thumb is instructed to move right and outside the circle while the other fingers remain stable (purple dots indicate original positions)

We created again two blocks of trials for each initial target position. Hence, we had 4 matching trials × 2 blocks × 8 shapes × 21 participants = 1344 data samples for shape matching and 2688 for moving.

Individual Finger Stability and Movement Range

To address the finger stability and independent finger control discussion items above, we created the following task consisting of two phases: In the first step of both phases, participants were given a target shape that they had to match in the conditions of Study 1 (i.e. non relaxed) and hold the posture for 2 seconds. This was to measure general finger stability in a totally still condition. The initial positions of all finger points were shown in purple so that participants could see how far they had deviated from it when they could not hold their fingers still. Then, in phase 1, they were instructed to move a specified finger as far as possible in given directions while maintaining the others still. For horizontal shapes, this direction was upwards then downwards and for vertical shapes rightwards and leftwards for non-thumb fingers. The thumb always had to move in all four directions in all cases. There were no speed requirements in this phase and participants were told to prioritise stability.

In phase 2, following initial shape matching, participants were told to move their fingers in the specified directions so that their projected points exited and immediately re-entered a circle around the initial position (Figure 6 right). This was to be done with relatively fast flicking motions mimicking air clicks. The radii of the threshold circles corresponded to the mean + 2 standard deviations of the displacement of that finger when it was supposed to remain still during movement of the instructed finger in phase 1. In other words, the less a

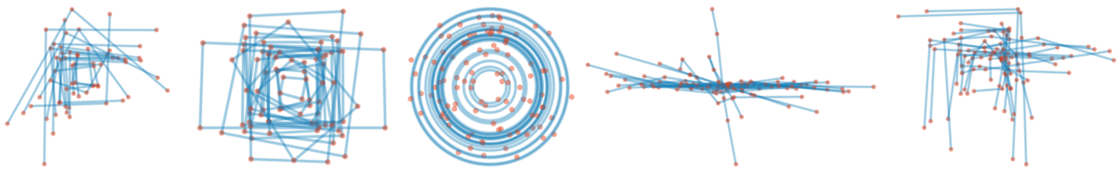


Figure 7. Merged centred view of participants' free shapes. From left to right: Short Corner, Rectangle, Circle, Line, Long Corner.

non-instructed finger was stable during movement of the instructed finger, the larger this threshold circle was.

Those measurements were performed for all shapes except the circle, which does not have specific finger position requirements and thus does not lend itself to this kind of task.

Results

Shape Elicitation

We combine the shapes produced using MCP in Study 1 with those of Study 2. We remove participants who performed Study 1 in the set of Study 2 to keep only unbiased data.

Figure 7 shows a merged view of the shapes formed by participants centred around their centroid. Except for *Circle*, where we fit the shape so as to better represent the differences, the projected dots are connected with each other so as to correspond to the instructed shape, but without any geometrical fitting. As is very apparent from the figure, most participants aligned shapes with the screen axes, especially the horizontal axis. Lines, in particular, were overwhelmingly formed parallel to the x-axis of the display. Corners were mostly formed in the H configuration. This justifies our study designs to consider only axis-aligned targets.

With regard to lengths, perimeters and circumferences, the means of the measured values for each shape are (in cm): Short Corner: 100 (sd=49), Rectangle: 192 (75), Circle: 209 (72), Line: 131 (45) and Long Corner: 125 (53). Those values are commensurate with the sizes of the target templates that we chose to use in our experiments.

Shape Matching and Moving

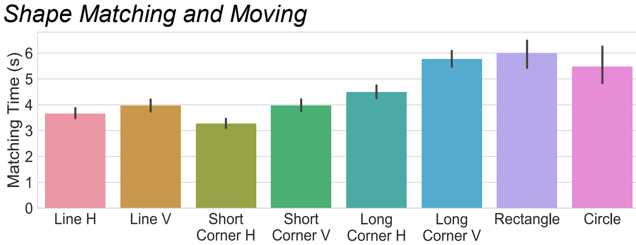


Figure 8. Mean matching times for each shape

Figure 8 shows the mean times for the shape-matching task (including dwell), again, after removing outliers corresponding to values that were more than three times the standard deviation away from the mean. The general pattern is similar to Study 1 (see Figure 5), except for *Circle* and *Rectangle*, which fare comparatively worse. *Short Corner* is the fastest shape at just above 3s.

We can essentially group the shapes in two categories, the first fast group (<5s) consisting of the *Short Corners*, the *Lines* and *Long Corner H* and the second slow group (>5s) consisting of *Long Corner V*, *Rectangle* and *Circle*. We confirm here as well that people perform better with horizontal shapes than with their vertical equivalents. *Long Corner V* is one of the slowest shapes again. The cases of *Circle* and to a lesser extent of *Rectangle* are interesting. *Circle* was among the faster shapes for MCP in Study 1, but adding the orientation requirement apparently made participants less efficient. We note, however, that the standard deviation, and thus variability, is considerably higher for those two shapes. This reflects our observation of some participants struggling with those shapes, while others were much more comfortable forming them. Producing lines and short corners was generally less problematic for everybody. The participants' subjective ratings in Figure 9 tend to show that *Circle* was not necessarily the least preferred shape, even though the differences with *Long Corner V* are not significant.

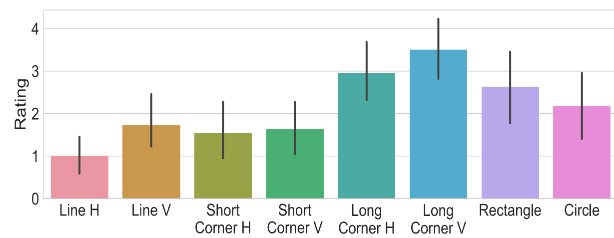


Figure 9. Participant ratings of how easy it was to form and move shapes on a scale from 1 to 7 (1=easiest, 7=least easy)

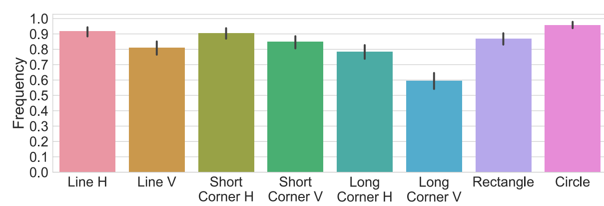


Figure 10. Frequency of shape preservation during motion.

As for shape stability during motion, we can see from Figure 10 that *Circle* is one of the shapes that participants were able to maintain the most often. We looked at the speed at which the shape was moved across the screen and we can confirm that the high stability was not due to participants being more careful. *Long Corner V* was the least stable of all shapes.

For shapes with fewer than five fingers, the distances of the closest finger not participating in the formation of the shape were, on average, above 1m for *Rectangle* and the *Lines* and above 2m for the *Short Corners*, albeit with high variability.

Individual Finger Stability and Movement Range

On average, maximum finger displacement in the posture-holding phase was between 4 and 7cm, with the highest values for thumb (6.6cm) and little finger (6.5cm).

Figure 11 shows the displacements of non-instructed fingers in the flicking condition. The measurements provide hints towards defining the size of activation zones out of which individual fingers move to trigger an action associated with a shape. If we define such zones to be circular, a radius of 16cm seems adequate for index and middle fingers and 10cm for the thumb. For Short *Corner H*, 12 cm for both index finger and thumb is a reasonable value. Ring and little fingers should be avoided as controls as they are the least individuated fingers. The lengths are of course dependent on the distance of the user to the screen. Lower values can be chosen for near-screen multiray, while lengths need to be increased for higher distances, with possible dynamic adjustments when the user moves towards or away from the display.

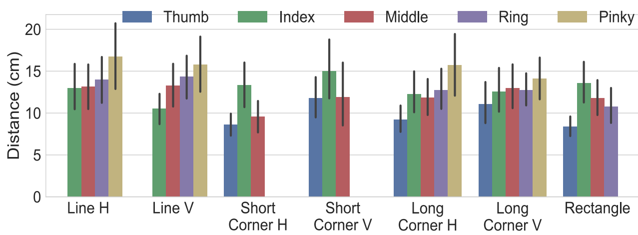


Figure 11. Finger displacement when flicking instructed finger

Participant Feedback

As in Study 1, seven participants reported that there was a mismatch between the physical hand postures and the projected points on the screen. The reason for this, as given by one of the participants, is that once a shape materialises on the screen, the brain tends to consider the finger arrangement as a whole instead of as a set of separate fingers. Consequently, it is sometimes confusing when the slight movement of a finger disrupts the entire shape. Participants, who adopted that mental model tended to rotate their entire hand in order to try to match the orientation of circles, although it was only determined by the middle finger. A similar phenomenon occurred with the long corner shapes, for which it was not immediately clear to participants that when the shape was not correctly formed it was mostly due to the index finger not being close to the right-angle point.

Three people stated that having to keep all finger points on the screen to form and maintain a shape was sometimes an issue. This problem is related to the size and limits of the display. Possibly multiray is more suitable for very large screen spaces perhaps extending to other surfaces in a room.

Finally, six participants said that the constraint to keep non-participating fingers far away affected their comfort, especially for lines and *Rectangle*. We therefore recommend using the shape-breaking technique described above only for shapes that require three and fewer fingers, i.e. *Short Corners* and two-finger interactions.

Discussion

Unsurprisingly, shapes that require fewer fingers (*Short Corners*) and fewer constraints were generally more efficient, but there were also results that we did not foresee. For instance, we did not expect differences between horizontal and vertical shapes to be so stark. Perhaps the biggest hurdle to absolute multiray is the discrepancy that can initially exist between the physical posture of the hand and the shape produced on screen. The need to coordinate and identify which fingers to move to form the desired shape is something that requires getting used to. This varies considerably between people, however, especially for shapes with more than four fingers. At one end of the spectrum, there was a participant who could virtually not produce any of the shapes in reasonable times. At the other end, people with experience with finger-gesturing interfaces and musicians were much more agile with multi-finger raycasting. Those participants were very quick at understanding the right finger formation to adopt in order to produce the desired result. They further commented that they enjoyed the experiments, hinting at possible applications of multiray for games, where mastering such techniques would be part of the challenge.

For regular applications, where training time needs to be minimised, we generally recommend favouring shapes with few fingers or horizontal shapes with few non-aligned fingers. To improve the performance of other shapes, the detection tolerance can be increased. This has the consequence that possibly false positives may increase, but the impact of those mistakes can be mitigated by limiting the number of shapes using the same number of fingers and by not assigning them to operations with high error costs.

APPLICATIONS

We now present a number of examples of how multiray interaction can be used in a set of demonstration applications. Most of those examples result from the findings of our experiments. In particular, except when used as free non-axis aligned shapes, we only adopt horizontal postures in our designs. Furthermore, we fine-tune our detection techniques so as to be slightly more tolerant for shapes that were more difficult to detect. Apart from those optimisations, we use the same setup and software base as in the experiments.

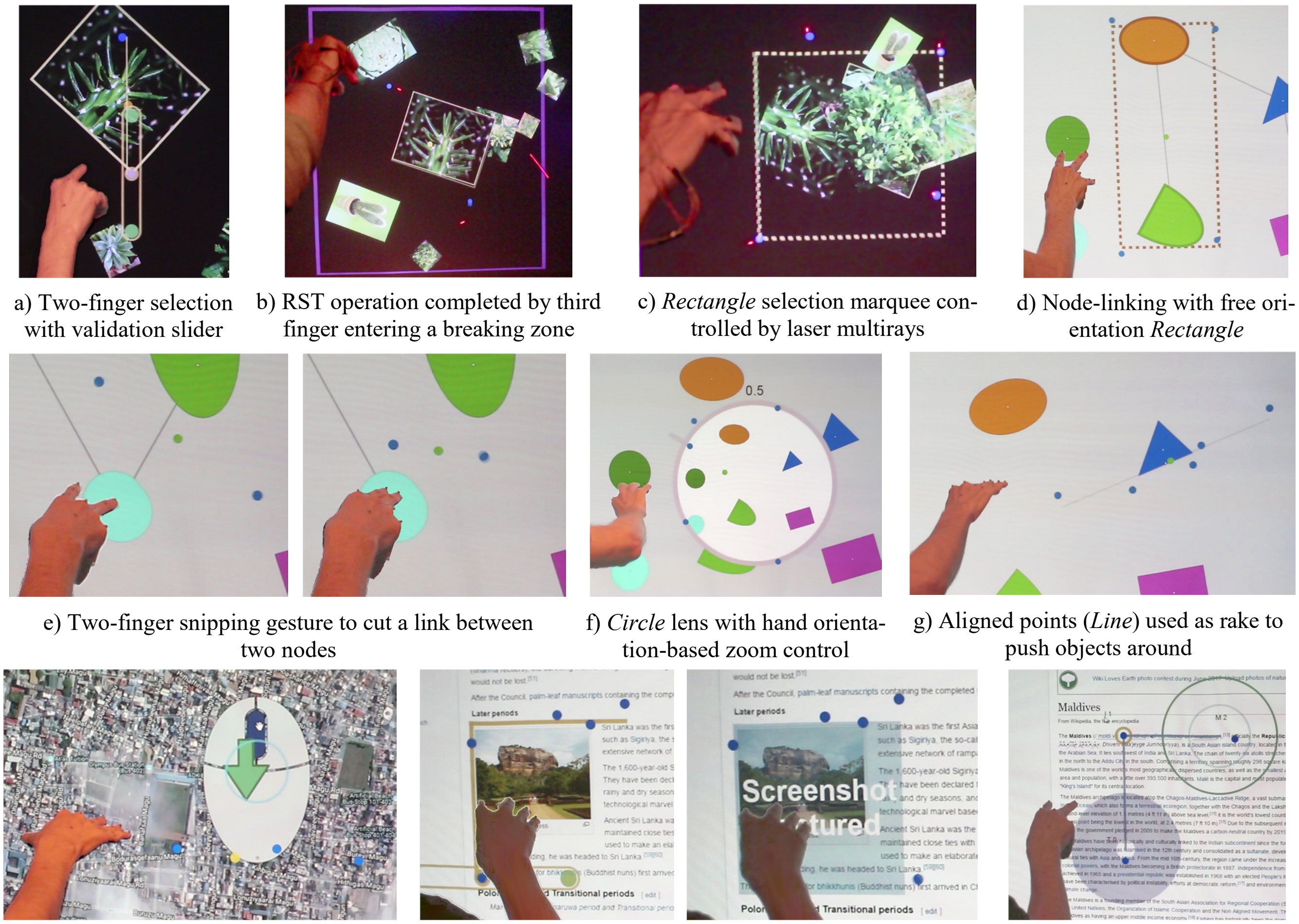
The purpose of the following applications is only to demonstrate possible mappings of multiray shapes to example manipulations. The prototypes are not meant to be feature-complete. We are also not suggesting that each of the below operations should ideally be realised with multirays.

Photo Manipulation Interface

This simple application consists of a set of photographs of various sizes and orientations scattered on the display.

Rotation Scale Translation

Classic rotation translation and scale (RST) operations can be performed with two rays similar to MultiPoint [4], but to be able to more easily target small objects, we use the centroid of the two points, as in Study 1, rather than requiring both points to be on the target. The initial selection and initiation of the RST operation can be performed with a long dwell, as in our studies. But since we have multiple cursors, we can use one of them to explicitly trigger the operation with a visual control. Figure 12a shows an example of a validation slider that appears at the location of the index finger



a) Two-finger selection with validation slider
b) RST operation completed by third finger entering a breaking zone
c) *Rectangle* selection marquee controlled by laser multirays
d) Node-linking with free orientation *Rectangle*
e) Two-finger snipping gesture to cut a link between two nodes
f) *Circle* lens with hand orientation-based zoom control
g) Aligned points (*Line*) used as rake to push objects around
h) Mouse-wheel control with movement of *Line H*
i) Selection and validation of a screen shot area with Long Corner H
j) Mouse control with *Short Corner H* on web document

Figure 12. Sample interactions demonstrating applications of multirays

(the most individuated finger) after a very short dwell time (e.g. 100ms). The finger needs to be moved outside the slider to unlock it and trigger the start of the RST operation with the element currently under the centroid point.

Operation Validation

To complete the operation, dwell could be used again, but that might be more error-prone than for selection, as people tend to slow down their movement, sometimes to near immobility, when trying to carefully position the object. As a more intentional method, we propose using the middle finger, which is not participating in the RST operation and kept out of range, to explicitly confirm the operation by entering a breaking zone, for example defined by a frame, as shown in Figure 12b. Our measurements in Study 2 suggest that the size of the longest diameter of such a rectangle should not exceed 1m. In practice, it is also important that users can always visualise where this zone is and therefore we recommend defining such zones well within the peripheral vision.

Rectangular Marquee Selection

To select multiple elements, the rectangle shape can be mapped to a selection marquee (Figure 12c), where the conditions to maintain the rectangle can be relaxed to a simple finger count after it has been registered, as argued in the discussion of Study 1. To confirm the selection, a validation slider controlled by the index finger can again be used. As an alternative to *Rectangle*, the more reliable *Short Corner* can be used, if it is not already assigned to another function. After contents have been selected, an operation can be selected from a marking menu or another widget.

2D Physics Engine

In this physics engine interface, users interact with rigid bodies represented by geometrical shapes. The objects can be moved, caused to collide with and be linked to each other. The goal of this prototype is mainly to demonstrate non axis-constrained shapes whose orientation, controlled by the user, plays an active role.

Object-moving

On touchscreens, the shape of the hand and multitouch techniques can be leveraged to align and push objects [11, 44, 49]. We extend that concept to multiray, where forming a line creates a virtual rake with which bodies can be gathered and pushed around (Figure 12g). The studies have shown that both horizontal and vertical lines are relatively easy to form and maintain during motion, so that triggering and dragging lines in any orientation should not pose any major problem.

Object-Linking and Unlinking

Objects can be linked with oriented selection rectangles with very relaxed detection conditions (considering it was one of the more challenging shapes). Specifically, users form a freely oriented rectangular shape around the nodes that they want to connect and dwell for 1s to validate the selection (Figure 12d). An alternative method (that we have not implemented) would be to use two fingers with each ray pointing at the nodes to connect. We have not evaluated the simultaneous targeting of two points with raycasting and we believe that would be an interesting avenue to explore in the future.

To unlink two nodes, we created a snipping gesture, which consists in positioning two spread fingers such that their projected points are on each side of the connecting edge and pulling the two fingers together so that each point crosses to the other side, thus severing the link (Figure 12e).

Zooming Lens

We implemented the example of the zooming lens introduced earlier. The lens is summoned by forming a circle, which, as we showed, has high motion stability and thus lends itself to such a widget that is frequently moved around to inspect different areas of the workspace. The scaling factor is mapped to the orientation of the lens determined by the middle finger (Figure 12f). While circle orientation caused some trouble for participants, the zooming factor of a lens is not a parameter that needs to be precisely set and so we think this is an adequate design choice.

Web Browser

Our third application example is a web browser, which integrates multiray-based mouse controls. We tested those interactions within a map interface and text-based web pages such as Wikipedia articles (Figure 12 h-j).

Mouse Simulation

To show how typical mouse-operated desktop applications can be manipulated on large displays *as is* with multirays, we created a mouse simulation widget associated with the most efficient shape: *Short Corner H* (Figure 12j). The mouse cursor is mapped to the right-angle point cast by the index finger, while the two other fingers (middle finger and thumb) are used to trigger button events by moving out and back into their activation zones (the two large circles). The size of the shape is fixed upon its detection and is moved around as a rigid body by the index finger cursor. The thumb and middle finger control the left and right mouse buttons respectively. When a finger point exits its activation zone, the corresponding mouse button is pressed. It is released when the finger point enters the circle again (to reduce errors, we use circles of different sizes for press and release). Hence, to perform a click, the finger is flicked so that its projected point rapidly moves out and back into the activation zone, as in the last task of Study 2. Panning actions based on mouse-dragging can be executed by moving the thumb out, moving the shape and returning the thumb inside the circle. This action can be repeated several times to clutch and pan. The widget is dismissed using the zone-breaking technique described above.

We also support mouse wheel control using *Line H*. After the line is formed, the user can "roll" the wheel by moving the line up or down (Figure 12 h). We map the movement of the line to discrete steps to correspond with the increments of the wheel and to make sure that breaking the line to dismiss the widget does not accidentally trigger an unwanted action.

Screen Capture

As an extra feature, we implemented a screen capture function associated with *Long Corner H*. After forming, moving and enclosing the desired portion of the screen with the shape, the user executes the screen capture with a horizontal slider linked to the thumb point (Figure 12i). The action can be cancelled by bringing the thumb point towards the long segment of the corner thus collapsing the shape. We acknowledge that given our study results, this interaction might not necessarily be a good design choice for a real application, but we just include it here as an example of a potential mapping for a *Long Corner*.

CONCLUSION AND FUTURE WORK

We presented multiray, or multi-finger raycasting to interact remotely with large displays. As motivated in the introduction, we believe multiray fills a hole in the raycasting design space similar to multitouch filling a hole relative to single touch. While we do not mean to also imply multiray is automatically a significant intuitive leap like multitouch, we hope our work shows that people can use multiray to increase expressivity compared to single ray.

We conducted two formative experiments to examine various properties of the technique. Our main findings are: 1) Raycasting from the entire finger (MCP to tip) is more efficient than from just the distal phalanx (DIP to tip). 2) There are signs that single-point targeting with multiple fingers can compensate for noise, but our results were not statistically significant. 3) Horizontal shapes are faster and preferred by people. 4) Short corners and lines are the most efficient shapes. 5) Circle and rectangles are generally less efficient but they also have a high variability among users. 6) Circles are easiest to maintain during motion. 7) People can be initially troubled by the fact that shapes produced on the screen do not always match the physical postures of their hand.

Based on those experiments, we created three applications demonstrating several multiray interactions for common operations. Our work paves the way towards future evaluations in context, with more shapes and possibly in coexistence with other non-ray gestures to address the shortcomings that we identified. A further avenue to explore would be to introduce biases towards particular shapes (thus departing from absolute raycasting) so as to facilitate their detection and the correspondence with the physical posture of the hand. Finally, as with any mid-air gesture or single-finger raycasting technique, fatigue is an important aspect that needs to be taken into consideration and rigorously assessed.

REFERENCES

1. https://glofx.com/product/glofx-lzr-laser-gloves-green/
2. F. Argelaguet, C. Andujar and R. Trueba. 2008. Overcoming eye-hand visibility mismatch in 3D pointing selection. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, ACM, 43-46. 10.1145/1450579.1450588
3. G. Bailly, R. Walter, J. Müller, T. Ning and E. Lecolinet. 2011. Comparing Free Hand Menu Techniques for Distant Displays Using Linear, Marking and Finger-Count Menus. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part II*, Springer Berlin Heidelberg, 248-262. 10.1007/978-3-642-23771-3_19
4. A. Banerjee, J. Burstyn, A. Girouard and R. Vertegaal. 2012. MultiPoint: Comparing laser and manual pointing as remote input in large display interactions. *International Journal of Human-Computer Studies*, *70*, 10 (2012/10/01/ 2012), 690-702.
5. A. Banerjee, J. Burstyn, A. Girouard and R. Vertegaal. 2011. Pointable: an in-air pointing technique to manipulate out-of-reach targets on tabletops. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ACM, 11-20. 10.1145/2076354.2076357
6. S. Bateman, R. L. Mandryk, C. Gutwin and R. Xiao. 2013. Analysis and comparison of target assistance techniques for relative ray-cast pointing. *International Journal of Human-Computer Studies*, *71*, 5 (2013/05/01/ 2013), 511-532.
7. M. A. Brown and W. Stuerzlinger. 2016. Exploring the Throughput Potential of In-Air Pointing. In *Proceedings, Part II, of the 18th International Conference on Human-Computer Interaction. Interaction Platforms and Techniques - Volume 9732*, Springer-Verlag New York, Inc., 13-24. 10.1007/978-3-319-39516-6_2
8. G. Casiez, N. Roussel and D. Vogel. 2012. 1 € filter: a simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2527-2530. 10.1145/2207676.2208639
9. A. Cockburn, P. Quinn, C. Gutwin, G. Ramos and J. Looser. 2011. Air pointing: Design and evaluation of spatial target acquisition with and without visual feedback. *International Journal of Human-Computer Studies*, *69*, 6 (2011/06/01/ 2011), 401-414.
10. W. Fikkert, P. van der Vet and A. Nijholt. 2010. Gestures in an Intelligent User Interface. In *Multimedia Interaction and Intelligent User Interfaces: Principles, Methods and Applications*, (2010), 215-242.
11. M. Frisch, S. Kleinau, R. Langner and R. Dachselt. 2011. Grids & guides: multi-touch layout and alignment tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1615-1618. 10.1145/1978942.1979177
12. E. Ghomi, S. Huot, O. Bau, M. Beaudouin-Lafon and W. E. Mackay. 2013. Arpège: learning multitouch chord gestures vocabularies. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*, ACM, 209-218. 10.1145/2512349.2512795
13. A. Goguey, M. Nancel, G. Casiez and D. Vogel. 2016. The Performance and Preference of Different Fingers and Chords for Pointing, Dragging, and Object Transformation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ACM, 4250-4261. 10.1145/2858036.2858194
14. V. Gracia-Ibáñez, M. Vergara and J. L. Sancho-Bru. 2016. Interdependency of the maximum range of flexion–extension of hand metacarpophalangeal joints. *Computer Methods in Biomechanics and Biomedical Engineering*, *19*, 16 (2016/12/09 2016), 1800-1807.
15. C. Häger-Ross and M. H. Schieber. 2000. Quantifying the Independence of Human Finger Movements: Comparisons of Digits, Hands, and Movement Frequencies. *The Journal of Neuroscience*, *20*, 22 (2000), 8542-8550.
16. F. Haque, M. Nancel and D. Vogel. 2015. Myopoint: Pointing and Clicking Using Forearm Mounted Electromyography and Inertial Motion Sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 3653-3656. 10.1145/2702123.2702133
17. C. Harrison, R. Xiao, J. Schwarz and S. E. Hudson. 2014. TouchTools: leveraging familiarity and skill with physical tools to augment touch interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2913-2916. 10.1145/2556288.2557012
18. R. Jota, M. A. Nacenta, J. A. Jorge, S. Carpendale and S. Greenberg. 2010. A comparison of ray pointing techniques for very large displays. In *Proceedings of Graphics Interface 2010*, Canadian Information Processing Society, 269-276.
19. W. A. König, J. Gerken, S. Dierdorf and H. Reiterer. 2009. Adaptive pointing: implicit gain adaptation for absolute pointing devices. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, ACM, 4171-4176. 10.1145/1520340.1520635
20. R. Kopper, D. A. Bowman, M. G. Silva and R. P. McMahan. 2010. A human motor behavior model for distal pointing tasks. *Int. J. Hum.-Comput. Stud.*, *68*, 10 (2010), 603-615.

21. R. Kopper, M. G. Silva, R. P. McMahan and D. A. Bowman. 2008. Increasing the Precision of Distant Pointing for Large High-Resolution Displays. *Technical Report*,
22. Y. Kou, Y. M. Kow and K. Cheng. 2015. Developing Intuitive Gestures for Spatial Interaction with Large Public Displays. In *Proceedings of the Third International Conference on Distributed, Ambient, and Pervasive Interactions - Volume 9189*, Springer-Verlag New York, Inc., 174-181. 10.1007/978-3-319-20804-6_16
23. G. J. Lepinski, T. Grossman and G. Fitzmaurice. 2010. The design and evaluation of multitouch marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2233-2242. 10.1145/1753326.1753663
24. Z. M. Li, S. Dun, D. A. Harkness and T. L. Brininger. 2004. Motion enslaving among multiple fingers of the human hand. *Motor control*, *8*, 1 (Jan 2004), 1-15.
25. M. Liu, M. Nancel and D. Vogel. 2015. Gunslinger: Subtle Arms-down Mid-air Interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, ACM, 63-71. 10.1145/2807442.2807489
26. X. Lou, A. X. Li, R. Peng and P. Hansen. 2016. Optimising Free Hand Selection in Large Displays by Adapting to User's Physical Movements. In *Proceedings of the 2016 Symposium on Spatial User Interaction*, ACM, 23-31. 10.1145/2983310.2985754
27. S. Mayer, K. Wolf, S. Schneegass and N. Henze. 2015. Modeling Distant Pointing for Compensating Systematic Displacements. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 4165-4168. 10.1145/2702123.2702332
28. V. McArthur, S. J. Castellucci and I. S. MacKenzie. 2009. An empirical comparison of "wiimote" gun attachments for pointing tasks. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*, ACM, 203-208. 10.1145/1570433.1570471
29. T. Moscovich and J. F. Hughes. 2006. Multi-finger cursor techniques. In *Proceedings of Graphics Interface 2006*, Canadian Information Processing Society, 1-7.
30. B. A. Myers, R. Bhatnagar, J. Nichols, C. H. Peck, D. Kong, R. Miller and A. C. Long. 2002. Interacting at a distance: measuring the performance of laser pointers and other devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 33-40. 10.1145/503376.503383
31. M. Nancel, J. Wagner, E. Pietriga, O. Chapuis and W. Mackay. 2011. Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 177-186. 10.1145/1978942.1978969
32. T. Ni, R. P. McMahan and D. A. Bowman. 2008. Tech-note: rapMenu: Remote Menu Selection Using Freehand Gestural Input. In *Proceedings of the 2008 IEEE Symposium on 3D User Interfaces*, IEEE Computer Society, 55-58. 10.1109/3dui.2008.4476592
33. M. Ortega and L. Nigay. 2009. AirMouse: Finger Gesture for 2D and 3D Interaction. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part II*, Springer-Verlag, 214-227. 10.1007/978-3-642-03658-3_28
34. K. Pietroszek, A. Kuzminykh, J. R. Wallace and E. Lank. 2014. Smartcasting: a discount 3D interaction technique for public displays. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: the Future of Design*, ACM, 119-128. 10.1145/2686612.2686629
35. O. Polacek, M. Klima, A. J. Sporka, P. Zak, M. Hradis, P. Zemcik and V. Prochazka. 2012. A comparative study on distant free-hand pointing. In *Proceedings of the 10th European Conference on Interactive TV and Video*, ACM, 139-142. 10.1145/2325616.2325644
36. I. Poupyrev, T. Ichikawa, S. Weghorst and M. Billinghurst. 1998. Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques. *Computer Graphics Forum*, *17*, 3 (1998), 41-52.
37. M. Schwaller and D. Lalanne. 2013. Pointing in the Air: Measuring the Effect of Hand Selection Strategies on Performance and Effort. In *Human Factors in Computing and Informatics: First International Conference, SouthCHI 2013, Maribor, Slovenia, July 1-3, 2013. Proceedings*, (2013), 732-747.
38. P. Song, W. B. Goh, W. Hutama, C.-W. Fu and X. Liu. 2012. A handle bar metaphor for virtual object manipulation with mid-air interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1297-1306. 10.1145/2207676.2208585
39. S. Sridhar, A. M. Feit, C. Theobalt and A. Oulasvirta. 2015. Investigating the Dexterity of Multi-Finger Input for Mid-Air Text Entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ACM, 3643-3652. 10.1145/2702123.2702136
40. J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, A. Topalian, E. Wood, S. Khamis, P. Kohli, S. Izadi, R. Banks, A. Fitzgibbon and J. Shotton. 2016. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and

correspondences. *ACM Trans. Graph.*, *35*, 4 (2016), 1-12.

41. D. Vogel and R. Balakrishnan. 2005. Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, ACM, 33-42. 10.1145/1095034.1095041
42. R. Walter, G. Bailly, N. Valkanova, Jörg Müller. 2014. Cuenesics: using mid-air gestures to select items on interactive public displays. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, ACM, 299-308. 10.1145/2628363.2628368
43. W. Westerman. 1999. Hand tracking, finger identification, and chordic manipulation on a multi-touch surface. *PhD Thesis*, University of Delaware
44. D. Wigdor, H. Benko, J. Pella, J. Lombardo and S. Williams. 2011. Rock & rails: extending multi-touch interactions with shape gestures to enable precise spatial manipulations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1581-1590. 10.1145/1978942.1979173
45. M. L. Wittorf and M. R. Jakobsen. 2016. Eliciting Mid-Air Gestures for Wall-Display Interaction. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, ACM, 1-4. 10.1145/2971485.2971503
46. M. Wu and R. Balakrishnan. 2003. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, ACM, 193-202. 10.1145/964696.964718
47. M. Wu, S. Chia, K. Ryall, C. Forlines and R. Balakrishnan. 2006. Gesture registration, relaxation, and reuse for multi-point direct-touch surfaces. In *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, 185-192. 10.1109/tabletop.2006.19
48. H. P. Wyss, R. Blach and M. Bues. 2006. iSith - Intersection-based Spatial Interaction for Two Hands. In *Proceedings of the IEEE conference on Virtual Reality*, IEEE Computer Society, 123. 10.1109/vr.2006.93
49. C. Xiang, A. D. Wilson, R. Balakrishnan, K. Hinckley and S. E. Hudson. 2008. ShapeTouch: Leveraging contact shape on interactive surfaces. In *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*, 129-136. 10.1109/TABLETOP.2008.4660195
50. S. Zhai. 1998. User performance in relation to 3D input device design. *SIGGRAPH Comput. Graph.*, *32*, 4 (1998), 50-54.