Gunslinger: Subtle Arms-down Mid-air Interaction

Mingyu Liu, Mathieu Nancel, Daniel Vogel
University of Waterloo, Canada
{mliu,mnancel,dvogel}@uwaterloo.ca

ABSTRACT
We describe Gunslinger, a mid-air interaction technique using barehand postures and gestures. Unlike past work, we explore a relaxed arms-down position with both hands interacting at the sides of the body. It features 'hand-cursor' feedback to communicate recognized hand posture, command mode and tracking quality; and a simple, but flexible hand posture recognizer. Although Gunslinger is suitable for many usage contexts, we focus on integrating mid-air gestures with large display touch input. We show how the Gunslinger form factor enables an interaction language that is equivalent, coherent, and compatible with large display touch input. A four-part study evaluates Midas Touch, posture recognition feedback, pointing and clicking, and general usability.

Author Keywords
wearable; gestures; barehand; touch; large displays

ACM Classification Keywords
H.5.2. Information interfaces (e.g., HCI): User Interfaces.

INTRODUCTION

Most mid-air barehand techniques use large hand and arm gestures performed in front of the body [3]. Arguably, this is partly due to limited tracking capabilities: large arm motions in front of the body are easier for sensors to track. The problem is that large motions can also be tiring [8], conspicuous, and require generous physical space to perform.

To make barehand gestures smaller, more comfortable, and more socially acceptable, they should be made more subtle, meaning "fine or delicate in meaning or intent." Subtle gestures require precise finger tracking with minimal occlusion, most easily achieved by mounting sensors on the body (such as fingers [5, 4], hands [11, 16, 10], arms [9], shoulder [7], chest [13], and shoes [1]). However, many of these tracking solutions require cumbersome or invasive hardware and the focus of most past work has not been on interaction subtlety.

Our work introduces Gunslinger, a mid-air barehand interaction technique using hand postures to trigger command modes, and small finger and hand movements for events and parameter control. Unlike past work, Gunslinger explores an 'arms down' body stance where both sets of fingers are tracked in mid-air with thigh-mounted sensors (Figure 1). This stance not only makes input more subtle, but two-handed input and the reduced physical space needed to perform gestures is also more compatible with touch input on large displays. For example, Gunslinger can be used exclusively from a distance or mixed with touch input when near a display. We show how this can be achieved with an input vocabulary that is equivalent, coherent, and compatible across mid-air and touch input modalities, partly realized with a Gunslinger-enabled touch hand inference technique. The results of a four part evaluation show Gunslinger has little Midas touch, reliable posture detection, good pointing throughput, and acceptable usability even compared to faster touch input.

In addition to exploring an arms-down mid-air interaction space, we provide specific contributions including: a novel 'hand-cursor' to communicate recognized hand posture, command mode, and tracking quality; a flexible posture recognizer; a touch hand inference technique; a representative map navigation interaction vocabulary using Gunslinger with touch displays; and a study evaluating resilience to Midas Touch, posture recognition quality with and without hand cursor feedback, distant pointing performance, and general usability for Gunslinger alone and when mixed with touch.

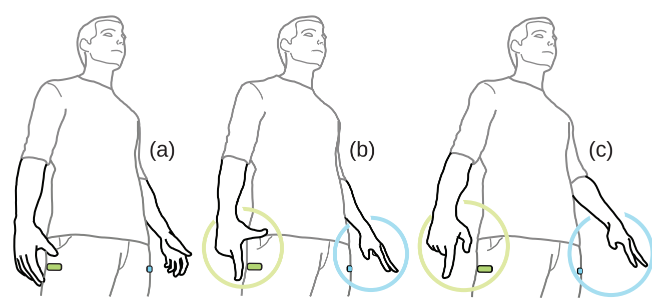


Figure 1: Gunslinger metaphor: (a) both hands down in neutral posture; (b) command modes triggered with hand postures, such as dominant hand thumb and index finger for pointing (green) and two fingers on the non-dominant hand for zooming (blue); (c) events or parameters are provided with finger movements, like folding the thumb down on the right hand to click or moving the two fingers to zoom in or out.

RELATED WORK

Arms-down interaction has been discussed in previous work [9], and some barehand mid-air input technologies could in theory be used arms down. Yet, there has been no exploration of a full interaction vocabulary performed from an arms-down stance explicitly focusing on subtlety.

Submitted to ACM UIST 2015

Environment-fixed Mid-air Input

Motion tracking systems using markers can reliably track fingers in a large space from a distance [19]. However, this level of environment-fixed tracking without markers remains challenging. While sensor capabilities will improve, tracking issues when hands are occluded by other body parts, other objects, or other people will not go away. The usual solution is to require people to perform large, explicit hand and arm gestures in front of their body to make tracking easier and reduce the chance of occlusion. The problem is that large motions are more tiring [8], socially conspicuous, and difficult to perform when very near a display.

Hand-mounted Mid-air Input

Sensors can be mounted onto, or near the fingers for more accurate tracking and to counteract occlusion. Gloves with sensors [11, 10, 16] can detect mid-air finger movements with no occlusion. Digits [9] mounts sensors on the inside of the forearm to enable precise finger tracking with arm orientation. Since the focus is sensing, arms-down interaction is briefly discussed but not evaluated and no specific arms-down vocabulary proposed. FingerPad [4] uses a nail-mounted magnetic tracker so one fingertip can be used over the others like a touchpad, while uTrack [5] uses a pair of magnetometers on the index finger and thumb to track fingertip movements in 3D. Mounting sensors on, or near fingers can be cumbersome, and while some technologies enable more subtle, arms-down input, this has not been an explicit goal. A related touch interaction example is PocketTouch [17], a technical proof-of-concept for a modified capacitive sensor placed in a pocket to enable arms-down touch interaction.

Body-mounted Bare-hand Input

Sensors can also be mounted on other parts of the body to track finger motions with minimal occlusion. Sixthsense [13] uses a chest mounted camera aided by color markers to detect hand and finger movements performed in front of the body. OmniTouch [7] uses a shoulder mounted depth camera to track finger positions on any nearby surface. ShoeSense [1] uses a shoe-mounted depth sensor to track hand motions from below, and can recognize some discreet postures like pinching, or the number of fingers when held horizontally in front of the body. The mounting point of sensors is an important consideration, and solutions like chest, shoe, or shoulder still force most gestures to be in front of the body.

Combining Mid-air Gestures with Touch

Another aspect is how to combine mid-air interactions with large touch-enabled large displays. Previous work has applied the principles of Proxemics, where the input possibilities change based on spatial factors like distance. For example, Vogel *et al.* [18] use mid-air gestures for mode selection from a distance, but change completely to touch input when near the display. Ballendat *et al.* restrict specific functionality to mid-air and touch based on a more complete set of proxemic relationships [2]. Bragdon *et al.* [3] combine mid-air, touch, mobile devices, and laptops together. Although some functionality is available across input modalities, Bragdon *et al.* explicitly state a design principle that "Each modality should have a separate use." Essentially, these are all traditional multimodal approached where each modality is dedicated to a specific function [15]. Our system treats mid-air gestures and touch more equally, so the most suitable input method can be used regardless of location.

GUNSLINGER

The Gunslinger name refers to the holster-like placement of the two 3D cameras and quick trigger postures like index-and-thumb pointing (Figure 1). This is reminiscent of cowboy 'gunslingers' drawing their guns in classic Hollywood films. The Gunslinger design follows five principles:

- *Relaxed* – Mid-air input should keep large muscles as relaxed as possible to reduce fatigue.
- *Precise and expressive* – Mid-air input should support a broad range of precise and controllable input tasks.
- *Always available* – Providing input should be possible without performing a universal input delimiter and without Midas Touch false-positives.
- *Display-focused* – The user should not have to look at their hands to understand system state or recognized responses.
- *Location independence* – Gesture articulation and sensing should be feasible regardless of nearby obstructions.

An arms-down posture satisfies *relaxed input* in terms of arm fatigue. Mounting 3D cameras on both thighs enables tracking *precise and expressive* finger movements and hand postures performed with both hands. Relaxed, natural postures (e.g. relaxed fist, open hand) are reserved for the neutral system state to avoid Midas Touch and commands are *always available* with a short transition to specific command hand postures. Additional feedback indicates the recognized hand posture, current command mode, and tracking bounds for both hands for *display-focused* input.

Hand Tracking from the Thighs

We mount a consumer Leap Motion (LM) device on each thigh just below the hips (Figure 2). The LM is a commercially available 3D camera with hand tracking software intended for desktop use. Our prototype tethers the LM devices to a desktop computer with long USB cables, but a wearable computer or Gunslinger-specific hardware would eliminate this. Hands and fingers are tracked with millimetre accuracy within a volume approximately .25 m^3 and software reports the size and orientation of the palm and fingers. The 3D cameras face out to enable high resolution finger tracking in a comfortable area when the arms are down, making input gestures feasible even near walls, other users, or displays for *location independence*. Since the cameras only sense hands when arms are down, many communicative gestures (e.g. waving, pointing) are explicitly ignored. We expect this further reduces Midas Touch.

Artificially Limited Control Area

The 3D cameras have a nearly hemispherical sensing area with radius approximately 0.6 m. Preliminary tests found that although the 3D cameras enable high-precision finger tip pointing with small wrist tilts, untrained users often used elbow and shoulder rotations to perform the same fingertip motions. Such movements would occasionally bring the user's hand outside the sensing range.

To encourage users to stay in the sensing range and adopt a relaxed posture, we limit input control to an area smaller than what the LM device supports. The artificially limited area is defined using a 12 cm radius disc located 15 cm away from the sensor along the z-axis (Figure 2). Gunslinger input is limited to palm positions projected along the y-axis that fall inside this disc. The size and location was tuned for a comfortable interaction range without shoulder strain, and the remaining sensor input space is used for out-of-control-area feedback. This limited control area also helps filter out input interference from nearby objects.

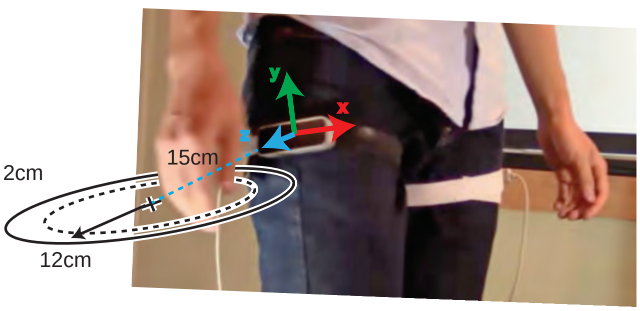


Figure 2: Leap Motion sensors are mounted on each thigh and an artificiality limited area is defined for Gunslinger input control. The area is defined using a 12 cm disc on the x-z plane centred 15 cm from the device. The dashed circle is the point where sensing limit feedback begins.

Hand Posture Recognizer

Gunslinger uses discrete hand postures to activate command modes. While maintaining a posture, subsequent hand or finger movements issue command events (e.g. clicking while pointing) and continuous or discrete command parameters (e.g. zoom level). Our postures are defined by which fingers are raised or folded and whether the thumb is stretched out, aligned with the palm, or tucked into the palm (Figure 3). Combining posture and movement creates a reasonably *precise and expressive* interaction language theoretically capable of $3 \times 2^4 = 48$ postures per hand. We focus on postures that can be formed in a *relaxed* and *display-focused* manner.

The LM software provides pre-computed features for palm position and digit tip positions in world space, and boolean flags indicating if each digit is raised or folded. Since the LM algorithms are tuned for desktop usage, we found additional heuristics necessary to compensate for misreported features: (1) each LM device tracks only the closest hand object per frame; (2) a hand detected at a distance greater than 60 mm, or continuously visible for less than 0.2 s is ignored to eliminate background noise and flickering; (3) extended fingers with tips less than 10 mm apart are collapsed into a single finger

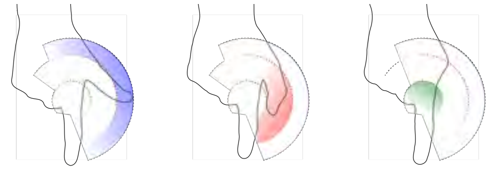


Figure 3: Thumb states: (left, in blue) thumb movement away from the hand, *thumb up* ; (middle, red) thumb movement near the hand, *thumb down*; (right, green) thumb tucked into the hand *thumb hidden*.

to handle the frequent occurrence of the LM reporting one finger as two fingers very close together.

Using these corrected features, we designed an efficient, simple, and generic finger posture recognizer in the form of a nearest-neighbour classifier. It uses a normalized similarity score s_i between the features of the current finger posture (C) and each finger posture in a vocabulary (V_i). Given C, two values are computed for each V_i: n_i is the absolute difference in the number of raised fingers; o_i is the distance, expressed in number of fingers, between the pattern of raised fingers in C and V_i. For example, if the Pinky is raised instead of the Index finger, then o_i = 3 (fingers away). In more complex situations where more than one digit is mismatched, o_i only considers the worst (i.e. most distant) mismatched digit. The following computes o_i given the extended state of the 4 fingers ($k = \{1,2,3,4\}$) in the vocabulary candidate posture ($d_{V_i,k} = \{0,1\}$) and current posture ($d_{C,k} = \{0,1\}$):

$$o_i = \max\left\{ \begin{array}{l} \forall j : d_{V_i,j} \neq d_{C,j}, \\ \min\left\{ \begin{array}{l} \forall k \neq j : d_{V_i,k} = d_{C,k} = 1 \\ \qquad \vee \begin{pmatrix} (d_{V_i,k} - d_{C,k}) = -(d_{V_i,j} - d_{C,j}) \\ \wedge\, d_{V_i,k} \neq d_{C,k} \end{pmatrix}, \\ |j-k| \end{array} \right\} \end{array} \right\}$$

In practice, o_i primarily detects when a digit is mistakenly raised instead of another and reinforces n_i when the numbers of raised digits do not match.

The similarity score between C and V_i is the weighted sum of these normalized values: $s_i = w_n \frac{n_i}{4} + w_o \frac{o_i}{4}$. w_n and w_o are constant weights set to .35 and .65. Lower values indicate better matches, so C is recognized as V_i if $S_i = \operatorname{argmin} S_i \mid S_i < \theta$. We found that a threshold of $\theta = .2$ provides accurate recognition without being overly restrictive.

Once the finger posture is determined, the thumb state is classified using the normalized distance between the thumb tip and the index metacarpophalangeal joint (i.e. knuckle). The maximum thumb distance is calculated in a short calibration step recording the thumb tip position in an open hand and a clenched fist. The thumb is classified as *up*, *down* or *hidden* (Figure 3) using distance thresholds of 75%, 50%, and 25% with additional transition hysteresis adjustment of 10%.

Visual Feedback

On-screen feedback visualizes how the system is classifying current hand postures, what commands (if any) are triggered, and notifying when either hand is nearly or completely out of sensing range. This follows our *display-focused* guideline and also helps people learn the interaction vocabulary itself.

Gunslinger accomplishes this by decorating two 'hand cursors' with simple graphics (Figures 4, 5, 6).

The cursors already serve as direct manipulation feedback (e.g. pointing location, selected menu item) making them a natural focus of attention. The dominant hand cursor also functions as a positional pointer and the non-dominant cursor is fixed near the bottom left of the display since it is associated with non-positional controls such as zoom and rate-controlled pan. For very large displays, the non-dominant cursor could follow the dominant one like a trailing widget [6] to minimize visual distance. The cursors and feedback are black and white with contrasting outlines to provide maximal contrast above any background image.

Hand posture feedback

We expect that people know how they hold their own hand through proprioception, but the way the system sees these postures might differ. For example, sensor errors, posture recognition thresholds, or misaligned frame of reference could cause recognition errors that might be easily corrected with slight adjustments in posture. We provide real-time, discreet visual feedback about how the LM device perceives hand postures in the form of a hand proxy ring, a stylized graphic of a hand surrounding the cursor (Figure 4). The ring has bumps representing raised or tucked fingers as perceived by the LM device. The thumb is rendered as bump that moves away from, or closer to the digits to communicate detected thumb posture: stretched out (Figure 4-a), aligned with the palm (4-b) and tucked (4-c). Depicting the user's hands on each cursor could also disambiguate hands in bimanual pointing configurations, such as scaling objects from two corners.

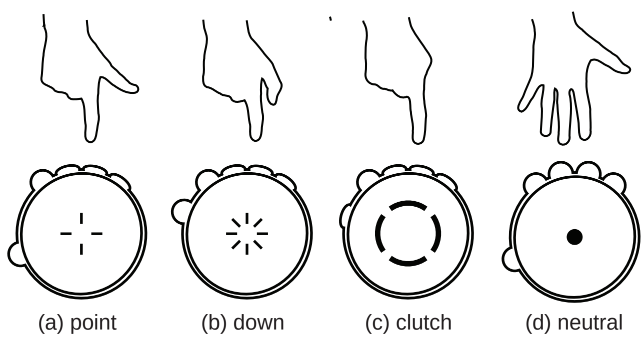


Figure 4: Hand cursor represents the recognized state of fingers and thumb as large or small bumps on a ring, and a centre icon represents active command state. For example: (a) pointing; (b) thumb click down; (c) thumb tuck clutch; (d) open hand neutral.

Command mode feedback

A central icon represents the cursor hotspot and the currently recognized command mode. For example, in our demonstration system the centre of the dominant hand cursor can display a four-branched aiming sight (Figure 4-a) for pointing, an eight-branched aiming sight (4-b) for pressed-down state when clicking and dragging, a dashed circle (4-c) for clutching, and a dot (4-d) when in the neutral state. Discrete sounds complement visual feedback, for example, we play subtle sounds for press-down and press-up. We used sound sparingly to limit annoyance and distraction.

Sensing limit feedback

The hand cursor also communicates when a hand is approaching, or has left, the artificially limited control area defined above. When the projected hand position is more than 80% from the disc centre (i.e. past dashed circle in Figure 2), two changes occur. First, cursor opacity decreases linearly from 100% to 20% corresponding to the outward 20% of the control range. The cursor never disappears for easy recovery. Second, in that same outward 20% range, the surrounding hand shape begins to shift from the centre icon in the opposite direction of the nearest range bounds. This animated offset makes the cursor feel like it is approaching the bounds and indicates the direction where tracking will improve. The centre icon does not shift to maintain direct manipulation feedback.

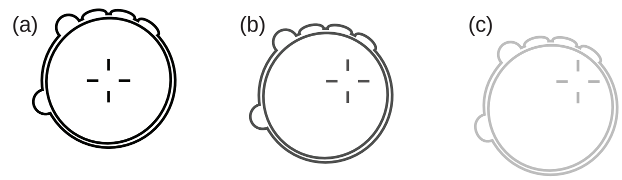


Figure 5: Opacity and shifting of the cursor ring to communicate proximity to control area bounds: (a) more than 20% from all bounds; (b) 15% from upper-right bound; (c) at or beyond upper-right bound.

INTERACTION VOCABULARY

As a proof of concept demonstration, we designed a Gunslinger interaction vocabulary for annotating and navigating a map on a large touch display (Figure 7). The vocabulary investigates a variety of control types (absolute/relative, direct/indirect, position/rate control), and shows how Gunslinger can be effectively combined with touch input.

Combining Gunslinger with Touch

As an extension to the five general Gunslinger design principles, we add three principles for combining Gunslinger with large touch displays. The goal is to minimize learning effort and enable free choice between input modalities.

- *Equivalence* – a common set of application functionality should be fully controllable with Gunslinger and touch (e.g. pointing using touch or using Gunslinger). This enables people to step back to get an overview and still accomplish the same tasks with Gunslinger.
- *Coherence* – Gunslinger and touch vocabularies should share morphological or semantic aspects. This can be external coherence (e.g. using established input conventions like two finger drag for scrolling) or articulation coherence such as mapping to the same hand (e.g. non-dominant navigates, dominant points with both Gunslinger and touch) or mapping to similar postures (e.g. two fingers opens a menu with both Gunslinger and touch). This helps transfer learning between mid-air and touch modalities.
- *Compatibility* – The requirements for space and tracking should support the simultaneous usage of Gunslinger and touch when close to the display (e.g. dominant hand points with touch while non-dominant navigates with Gunslinger, or vice-versa). This allows the combination of mid-air and touch to accelerate tasks (e.g. non-dominant Gunslinger

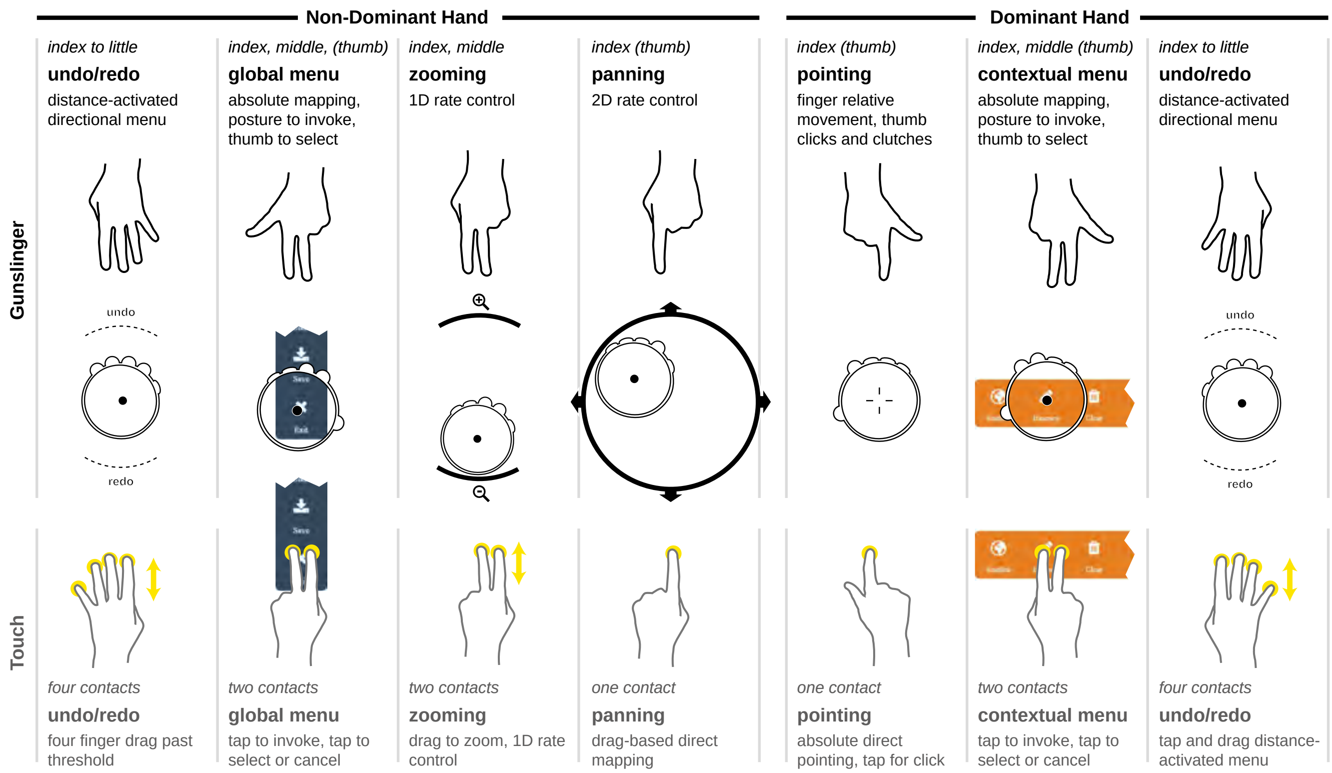


Figure 6: Gunslinger and touch interaction vocabulary for the map navigation demonstration system.

navigation with dominant touch-based object manipulation) or complete difficult tasks (e.g. dominant points with Gunslinger to reach distant targets while non-dominant navigates using touch).

Map Navigation Vocabulary

The Gunslinger and touch interaction vocabulary enables panning and zooming, selecting map style, defining landmarks through pointing, calculating an itinerary between these landmarks, and defining a zone within these landmarks (illustrated in Figure 6 and accompanying video). Any hand posture not included is an inactive neutral state. The most common postures to form naturally, the open hand and closed fist, are neutral. This vocabulary is designed to provide *equivalent* functionality between Gunslinger and touch.

Hand and posture mappings are designed for *coherence* across modalities. The number of fingers for a Gunslinger posture and number of multi-touch contacts both map to the same functionality: one for pointing and panning, two for invoking menus and zooming, four to undo-redo. The hand used also has coherent mappings: the dominant hand edits in context (pointing, contextual menu) while the non-dominant hand sets that context (pan-and-zoom, general menu). There is also external coherence: two-finger postures and two contacts invoke menus like the established two finger tap gesture, and two-finger postures and two contacts trigger zooming reminiscent of two finger zooming in Google Maps.

We adapted the vocabulary to match inherent differences between Gunslinger and touch input. The arms-down stance for Gunslinger requires an indirect mapping in a small operating range so rate or relative control are more suitable for continuous input. While the touch vocabulary can use standard absolute-direct mappings for pointing, panning, and zooming, Gunslinger uses rate-based control for clutch-free panning and zooming and relative control with clutching for high precision pointing. While the touch vocabulary can use surface contact to 'click' on a location or menu item, Gunslinger requires an explicit delimiter: thumb movements are used to click and clutch when pointing or selecting from menu items.

For *compatibility* when near the display, there are no bimanual-dependent mappings: each hand triggers and controls actions separately, and no command requires a combination of both hands. Postures can be combined if desired, like navigating with the non-dominant hand and pointing with the dominant hand. This allows modalities to be mixed: one hand can be used with Gunslinger and the other with touch. Since undo and redo are frequent commands, they are mapped to a four finger posture and a four finger touch performed with either hand for maximum convenience.

Touch Hand Inference

The vocabulary relies on discriminating between right and left hands. This is trivial with Gunslinger given the one-to-one mapping between hands and LM devices, but current touch displays do not identify which hand is used. We created a simple state-machine that uses Gunslinger input history, touch proximity, front facing stance, and user handedness to infer which hand was used to touch (Figure 7). The state machine implements these high level behaviours: if a touch starts while one LM devices detects a hand, then the

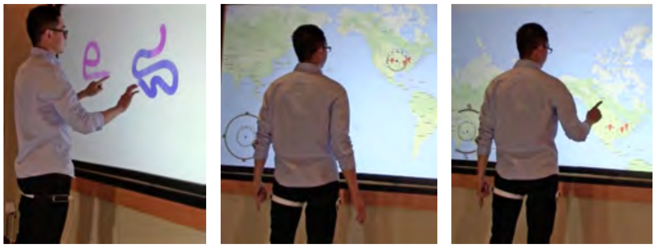


Figure 7: Left: Example of touch hand-discrimination. Middle: Bi-manual Gunslinger. Right: Bi-manual mixed touch+Gunslinger.

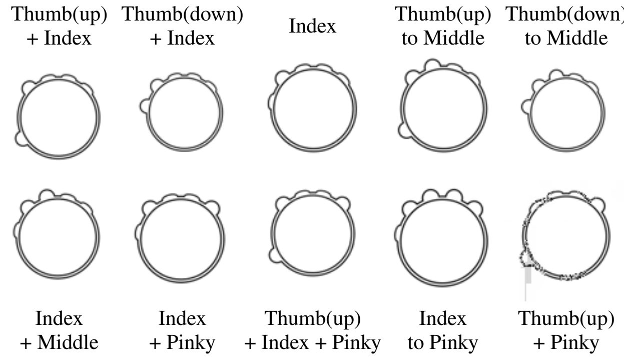


Figure 8: The 10 postures used in the posture recognition part.

touch is credited to the other hand; if a touch starts while no LM device detects a hand, the handedness of the user and the distance to existing touch points are used to guess which hand is used; if a new touch is far left of a current left touch, the new touch is labelled as left and the current touch is relabelled as right (the rule is inverted for right touches); if touch points associated to one hand move too far apart, distant points are reassigned as different hands. These rules do not provide perfect detection, but they work for common usage patterns. Fortunately, detection is easily correctable by swiping a hand past the thigh to reset touch-to-hand assignments. The full state machine is provided in the Appendix.

EVALUATION

Our goal is to validate technical and usability aspects of Gunslinger. Low arm fatigue [8] and social acceptance are a consequence of arms-down subtle interaction. The study is a sequence of experiments in four parts: (1) Midas Touch robustness; (2) effectiveness of posture recognition and hand cursor feedback; (3) arms-down pointing performance; and (4) general usability with and without touch. By completing these parts in sequence, participants incrementally learned and practised the system leading to the final usability part. The task, design, and results for each part are described individually below.

Participants and Apparatus

10 participants (3 female, mean age 24.2) completed the study. 4 had experience with mid-air game controllers and all were right-handed. We use the Gunslinger system described above with an 80", 1280 × 720 px, back projected display with a PQ Labs multi-touch overlay. All software is JavaScript HTML5 web applications, using the Google Maps API for the last part.

Part 1: Midas Touch

The goal of this part was to elicit "normal" conversational gestures in order to investigate whether these create false-positive gestures with Gunslinger. The LM devices were strapped on each thigh, then the participant's attention was diverted with a demographics interview conducted while they stood. The interview was extended with open questions such as "*List all the touch interfaces that you have ever used*" and additional follow-up questions. The system logged recognized postures throughout, but no feedback was displayed.

Results

The interview took 4.19 min on average (SD 43.8 s). Due to a technical issue, only right-hand postures were logged for this part.[1] On average, 5 different postures were recognized per participant (SD 3.3). The hand was outside the sensing range 89.7% of the time since conversational gesticulation often occurred in front of the body. Postures reserved for neutral states were recognized in most of the remaining time (fist 4.5%, open hand 1.5%). Vocabulary postures were detected the remaining 4.3% of the time.

Participants' hands were out of the LM device sensing range most of the time, and we detected few false positives even though participants were not trained and made no compensating behaviour. The lower quartile of the durations of detected postures is 267 ms, suggesting an activation duration filter could further reduce false positives. To test this, we reprocessed the raw logs with a 250 ms duration threshold and a more conservative $\theta = .15$ in our posture recognizer. This lowered the false-positive vocabulary postures to only 0.68% of the time. This is a worst-case result with users untrained and unaware of the technique. Training, as well as a more purpose-built sensing device would likely decrease false positives further. Overall, this suggests that using discrete arms-down postures to trigger interaction is a reasonable approach.

Part 2: Posture Recognition and Feedback

The goal of this part was to assess how well hand cursor feedback assisted posture formation, and how well posture recognition functioned. After being briefed on hand cursor feedback, participants completed a sequence of trials with each hand. Each trial began when they formed an open-hand neutral posture, then one of 10 target hand postures was displayed as a hand cursor (Figure 8). The participant formed the posture and held it for a defined time.

There were two conditions: *Feedback* and *No-Feedback*. In *Feedback*, the recognized pose was provided in real time with a second hand cursor. The correct posture had to be held for 500 ms (shown as a progress bar), with the time reset if an incorrect posture was detected. This served as a best-case scenario for posture recognition, since users can compensate for recognition errors by actively monitoring feedback. In *No Feedback*, participants had exactly 4 seconds (shown as a progress bar) to form and hold the posture to the best of their

[1] We later re-ran this part with 10 new participants and results were similar: left hand (90.7% of the time outside sensing, 3.7% neutral gestures, 5.6% vocabulary gestures); right hand (92.0% of the time outside sensing, 2.3% neutral gestures, 5.7% vocabulary gestures).

ability without feedback. The time did not reset if the wrong posture was formed. This served as a worst-case scenario where participants divert their full attention away from posture feedback. No Feedback always followed Feedback for each hand so participants had experience forming postures before feedback was removed. Hand order was counterbalanced and posture order was randomized. For each hand and condition, all postures were repeated 3 times. In summary: 2 feedback conditions × 2 hands × 10 postures × 3 replications = 120 data points per participant.

Results

Failure – No trials could fail with Feedback, but with No Feedback, the correct posture was not recognized in 82 trials (13.7 %). Posture had a significant effect on failure ($F_{9,81} = 6.31$, $p < 0.0001$). Index+Middle had no failures, a Tukey post-hoc test revealed it caused significantly less failures than any three-fingered posture, or any posture involving the Pinky. Hand had no significant effect on failure. Using the $\theta = .15$ and 250-ms filter mentioned above would only increase the number of failed trials by 4 in the No Feedback condition.

Completion time – This is the time taken to form the correct posture since trial start. After removing failed trials, we found significant effects of posture ($F_{9,59.47} = 5.88$, $p < 0.0001$), feedback ($F_{1,8.95} = 43.76$, $p < 0.0001$), and feedback × posture ($F_{9,58.19} = 3.19$, $p = .0033$) on completion time. Feedback was significantly slower than No Feedback (2.51 vs 1.44 s) indicating that although feedback decreases errors, it also slows people down. Tukey tests revealed that the index posture (1.30 s) was significantly faster than thumb+index+pinky (2.55 s), thumb(up)-to-middle (2.83 s) and index+little (3.07 s). Hand had no significant effect.

Part 3: Arms-down Pointing

The goal of this part was to evaluate pointing and clicking performance. The task is similar to previous studies [19]. Participants stood 2 metres from the display and acquired a sequence of circular targets using the point, clutch, and click postures. A trial was successful if the first click-down and click-up events occurred inside the target bounds. Each target had to be successfully selected to continue. We combined two Amplitudes (1400 and 350 mm) and three target Widths (40, 80 and 160 mm) creating an Index of Difficulty (ID) range of 1.7 to 5.2 bits. We used the gain transfer function and calibration process described in [14]. Participants completed 1 block of practice trials and 3 blocks of measured trials. Each block contained all combinations of Amplitudes and Widths with two repetitions: 3 Blocks × 3 Sections × 2 Amplitudes × 3 Widths × 2 repetitions = 108 data points per participant.

Results

We calculated error rate and median target acquisition time (the median accounts for skewed distributions). A multi-way Anova found a significant effect of Width on Error ($F_{2,18} = 23.53$, $p < 0.0001$): 40 mm (18.1 %) caused significantly more errors than 80 mm (8.3 %) and 160 mm (6.4 %). Errors were removed from time analyses. Times ranged from 1.62 s for ID 1.7 up to 3.17 s for ID 5.2. A Fitts' law regression has good fitness and indicates reasonable throughput: $497 + 483 \times ID$, $R^2 = .94$.

Part 4: Usability

The goal of the final part is to evaluate Gunslinger with realistic tasks in two phases, *Controlled* and *Open-Ended*.

After the full vocabulary was described (Figure 6), participants performed a sequence of three *Controlled* Tasks: T1 required locating and pinning two cities, each city must roughly fill the whole screen when pinning; T2 required undoing, then redoing the last pin-drop; and T3 required changing to satellite view, generating an itinerary using the contextual menu, then saving the itinerary with the global menu. Each task sequence was completed under three input conditions: *Gunslinger-only*, *Touch-only*, and *Mixed* in which the dominant hand uses touch and the non-dominant hand uses Gunslinger. The symmetric mixed configuration (dominant Gunslinger and non-dominant touch) was not included since it is likely only advantageous when reaching far targets on very large displays. The task order was fixed and input condition was counterbalanced. For each input condition, all three tasks were performed once as practice and a second time for observation. We logged task completion time and participants were asked to 'think-aloud.' After all conditions were completed, participants rated input condition for easiness, fatigue, speed, precision, and general opinion on a 7-point numeric scale (higher is better).

In the second phase, participants were given an *Open-Ended* task: "*You have a 3-month vacation with unlimited budget: plan your ideal trip, in the order that suits you best; generate an itinerary, and save it. You can use any combination of modalities that you like and take as much time as you need.*" The task requires the global and contextual menus and is designed to elicit map navigation and exploration. There was no minimum or maximum time limit, it was up to the participant to determine when they completed the task. We logged all input and participants were asked to 'think-aloud,' especially regarding input choice. This provided unconstrained subjective and observational feedback of general usability. At the end of this phase, participants were asked for additional comments.

Results

Controlled – We found a significant effect of input on completion time for T1 ($F_{2,18} = 4.09$, $p = .0344$), T2 ($F_{2,18} = 14.57$, $p = .0002$), and T3 ($F_{2,18} = 4.82$, $p = .0211$). Post-hoc tests showed Gunslinger (92.4 s) was significantly slower than Touch (60.6 s) for T1; Gunslinger (16.3 s) significantly slower than Mixed (11.7 s) and Touch (9.3 s) for T2; and Gunslinger (37.3 s) significantly slower than Mixed (16.5 s) and Touch (15 s) for T3. Participants said Gunslinger and Mixed were not as easy to use as Touch (medians 4 and 4.5 vs 6); 2 ratings were below neutral for Gunslinger, 1 for Touch, and 1 for Mixed. Overall, fatigue was not an issue (medians 5, 6, 5 for Gunslinger, Mixed, Touch) though 2 participants were below neutral for Gunslinger and 1 for Mixed. There may be some bias towards touch given experience and familiarity. Overall, perceived speed was comparable (medians 4.5, 5 and 5 for Gunslinger, Mixed, Touch), though 2 participants were below neutral for Gunslinger, 1 for Mixed, and 1 for Touch. Overall, precision was good (medians 5, 5, 6 for Gunslinger, Mixed, Touch), though 1 rated Gunslinger below neutral, and 1 for Touch. The general impression was good overall (medians 5, 5.5, 6 for Gunslinger, Mixed, Touch), though 2 rated Gunslinger below neutral, and 1 for Mixed.

Open-ended – Gunslinger alone was used mostly for saving (4 participants) and undo/redo (3 participants, 4 did not undo), marginally for navigation (1 for pan and zoom), and never for adding markers and computing itineraries. Mixed was used more often: for adding markers (3), for panning and zooming (5 and 6), and for computing itinerary (1). Participant comments provided interesting insights. The novelty of Gunslinger was noted ("refreshing", "Touch is boring. I like [Gunslinger] more", P5 and P10), but also that Gunslinger may have hindered performance (P10) especially compared to Touch (P4, P7). Comments about fatigue favoured Gunslinger, with statements saying it was more relaxing than touch (P10, P9). Feelings were mixed about Gunslinger pointing, some had comments like "intuitive and subtle" (P9) others found it impractical (P5). Mixed was appreciated as a sensible (P1, P9) and "more natural" (P6) combination, but requiring more practice (P4). Gunslinger was considered advantageous at a distance with larger displays (P2), and up close with high targets (P9). Some said Gunslinger had adequate feedback (P3) and was "quite responsive" (P9), but some also said Gunslinger is "too sensitive" (P3) and the sensing range is too small (P6, P9).

Discussion

Overall, Gunslinger is usable with acceptable performance. Arms-down postures are promising: Midas touch is minimal and 7 to 10 postures of various complexities can be performed and recognized reliably, even without visual feedback. Using a lower posture recognition threshold and introducing a 250 ms detection window will further reduce false positives. When designing future vocabularies, postures that take longer to form should be reserved for infrequent commands.

Arms-down pointing and clicking is achieved with reasonable time and error rate, despite the novelty of the technique and of its unusual stance. All map tasks were feasible with Gunslinger and although most participants did not perceive a pronounced speed difference, task completion times with Gunslinger are slower than touch. This result is not surprising: touch is more familiar, has very high quality tracking, and tactile feedback – this is hard to beat. However, this does not nullify Gunslinger's usefulness: direct touch is not viable at a distance and the performance and enthusiasm for mixing Gunslinger and touch is encouraging. The majority of participants also said Gunslinger was less tiring, perhaps speed alone is not the definitive measure. Furthermore, Gunslinger can be combined with touch interaction both physically and semantically without causing high fatigue.

There are opportunities for improvement, for example comments indicating that Gunslinger may be "too sensitive". We believe this is partly due to occasional erroneous input when a finger providing continuous control (e.g. pointing) sends a short, high-speed "jerk" as it curls in to form a neutral fist posture. Such erroneous movements could be detected and automatically corrected.

CONCLUSION AND FUTURE WORK

We introduced Gunslinger, a mid-air barehand interaction technique using arms-down, subtle input to reduce physical input space, fatigue, and social awkwardness without sacrificing expressiveness. The technique features a new kind of hand-cursor feedback, to show recognized hand posture, command mode, and tracking quality, and is implemented with a simple, but flexible hand posture recognizer. An implemented interaction vocabulary for map navigation demonstrates how Gunslinger can be combined with touch input, This is supported by a touch hand inference method leveraging the arms-down form factor. The results of a four-part evaluation validate the technical feasibility and usability of the Gunslinger approach.

As future work, we plan to explore: using Gunslinger while seated; using Gunslinger for collaborative interactions, such as moving a hand from another user's LM device to one's own to transfer data; using Gunslinger for mid-air text entry by extending a technique like Vulture [12]; and designing a deployable Gunslinger-specific device (focusing on size, portability, resilience to outside light sources, and additional sensors such as sonar to further reduce interference). Finally, although our focus has been on the large touch display context, we imagine using Gunslinger with head-mounted displays for virtual and augmented reality, or for controlling a smartphone when in a pocket or bag.

ACKNOWLEDGEMENTS

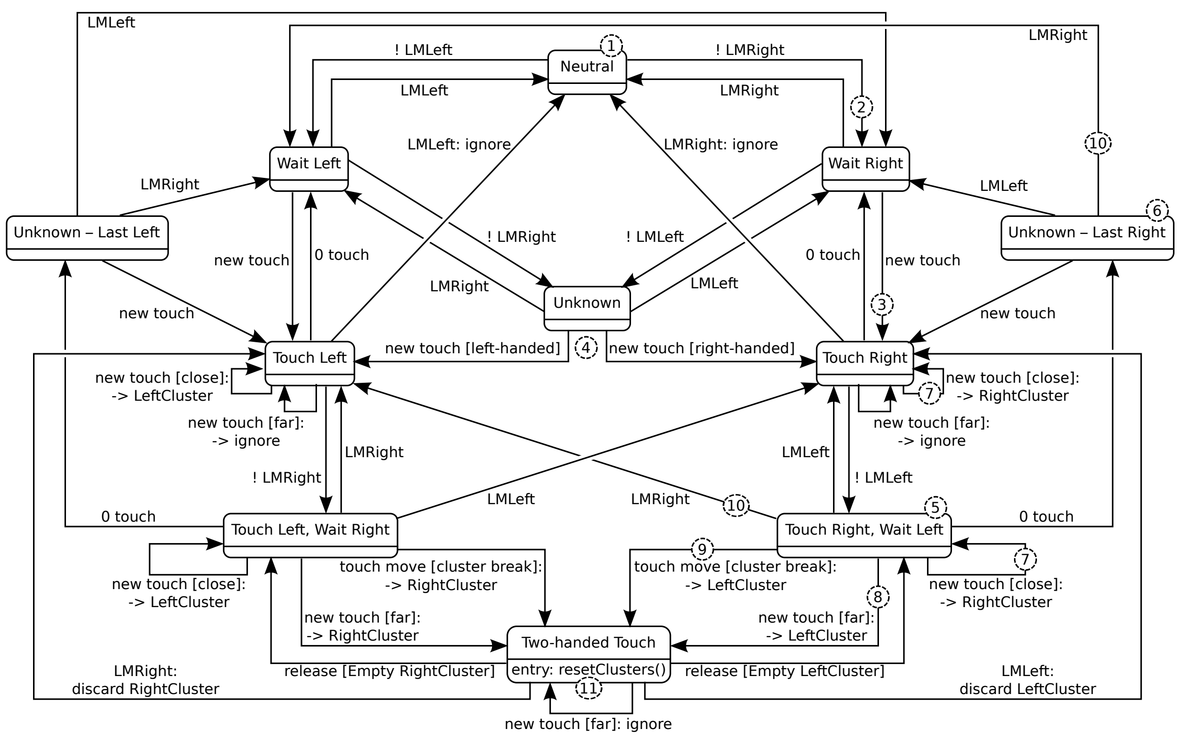
Funded in part by NSERC Discovery Grant 402467

REFERENCES

1. Bailly, G., et al. Shoesense: A new perspective on gestural interaction and wearable applications. In *Proc. CHI '12*, ACM (2012), 1239–1248.
2. Ballendat, T., Marquardt, N., and Greenberg, S. Proxemic interaction: Designing for a proximity and orientation-aware environment. In *Proc. ITS '10*, ACM Press (2010).
3. Bragdon, A., DeLine, R., Hinckley, K., and Morris, M. R. Code space: Touch + air gesture hybrid interactions for supporting developer meetings. In *Proc. ITS '11*, ACM (2011), 212–221.
4. Chan, L., et al. Fingerpad: Private and subtle interaction using fingertips. In *Proc. UIST '13*, ACM (2013), 255–260.
5. Chen, K.-Y., Lyons, K., White, S., and Patel, S. utrack: 3d input using two magnetic sensors. In *Proc. UIST '13*, ACM (2013), 237–244.
6. Forlines, C., Vogel, D., and Balakrishnan, R. HybridPointing: fluid switching between absolute and relative pointing with a direct input device. In *Proc. UIST'06*, ACM (2006), 211–220.
7. Harrison, C., Benko, H., and Wilson, A. D. Omnitouch: Wearable multitouch interaction everywhere. In *Proc. UIST '11*, ACM (2011), 441–450.

8. Hincapie-Ramos, J. D., Guo, X., Moghadasian, P., and Irani, P. Consumed endurance: A metric to quantify arm fatigue of mid-air interactions. In *Proc. CHI '14*, ACM (2014), 1063–1072.
9. Kim, D., et al. Digits: Freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proc. UIST '12*, ACM (2012), 167–176.
10. Levesque, J.-C., Laurendeau, D., and Mokhtari, M. Bimanual gestural interface for virtual environments. In *Proc. VR '11* (March 2011), 223–224.
11. Lu, G., Shark, L.-K., Hall, G., and Zeshan, U. Immersive manipulation of virtual objects through glove-based hand gesture interaction. *Virtual Real. 16*, 3 (Sept. 2012), 243–252.
12. Markussen, A., Jakobsen, M. R., and Hornbæk, K. Vulture: A Mid-air Word-gesture Keyboard. In *Proc. CHI '14*, ACM (2014), 1073–1082.
13. Mistry, P., and Maes, P. Sixthsense: A wearable gestural interface. In *Proc. SIGGRAPH ASIA '09*, ACM (2009), 85–85.
14. Nancel, M., et al. High-precision pointing on large wall displays using small handheld devices. In *Proc. CHI '13*, ACM (2013), 831–840.
15. Oviatt, S. Ten myths of multimodal interaction. *Commun. ACM 42*, 11 (Nov. 1999), 74–81.
16. Piekarski, W., and Smith, R. Robust gloves for 3d interaction in mobile outdoor ar environments. In *Proc. ISMAR '06*, IEEE Computer Society (2006), 251–252.
17. Saponas, T. S., Harrison, C., and Benko, H. PocketTouch: Through-fabric Capacitive Touch Input. In *Proc. UIST '11*, ACM (2011), 303–308.
18. Vogel, D., and Balakrishnan, R. Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users. In *Proc. UIST '04*, ACM (2004), 137–146.
19. Vogel, D., and Balakrishnan, R. Distant freehand pointing and clicking on very large, high resolution displays. In *Proc. UIST '05*, ACM (2005), 33–42.

APPENDIX: SINGLE-USER HAND INFERENCE STATE-MACHINE



Events are represented with the format event [condition]: action.

Main features: (1) Neutral state: both LeapMotion devices (LMs) detect a hand (events LMLeft, LMRight) and no touch is detected on screen. (2) A hand disappears from one of the LMs (!LMRight or !LMLeft), but no touch detected yet: expecting that hand. (3) Simple touch case: a touch is detected while only one LM detects a hand. (4) Both LMs detect no hand and no touch detected: next touch is expected from dominant hand. (5) One hand is touching the screen, the other one disappeard from the other LM. (6) Unknown state with knowledge of the hand that performed the last touch: the next touch will be on the same side unless LM events indicate otherwise. (7) Close touches are assigned (->) into clusters corresponding to the fingers of a same hand (resp. ClusterLeft and ClusterRight) if [close] enough. Touches that appear (8) or move (9) [far] from an existing cluster are assigned to the other cluster if possible. (10) In cases where a LM event challenges the current state (e.g. the right hand is thought to be on the screen, but the right LM detects a hand), the state is corrected accordingly. This enables users to correct wrong inferences. (11) Both hands are on the screen: the clusters are reset depending on their relative locations.