MatchPoint: Spontaneous Spatial Coupling of Body Movement for Touchless Pointing

Christopher Clarke
Lancaster University
Lancaster LA1 4WA, U.K.
c.clarke1@lancaster.ac.uk

Hans Gellersen
Lancaster University
Lancaster LA1 4WA, U.K.
hwg@comp.lancs.ac.uk

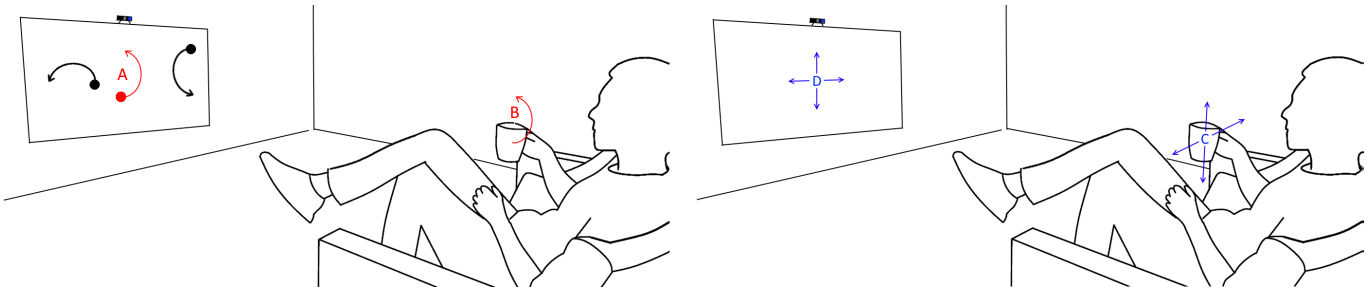

Figure 1. Spontaneous spatial coupling is a hybrid technique of motion-matching and pointing. Controls in the form of moving targets are presented to the user (A). When the user synchronises their movement with a target (B), a spatial coupling is created between the input modality (C) and the control (D). The technique enables ad hoc appropriation of any part of their body, or any object they hold, as a pointing device.

ABSTRACT

Pointing is a fundamental interaction technique where user movement is translated to spatial input on a display. Conventionally, this is based on a rigid configuration of a display coupled with a pointing device that determines the types of movement that can be sensed, and the specific ways users can affect pointer input. *Spontaneous spatial coupling* is a novel input technique that instead allows any body movement, or movement of tangible objects, to be appropriated for touchless pointing on an ad hoc basis. Pointer acquisition is facilitated by the display presenting graphical objects in motion, to which users can synchronise to define a temporary spatial coupling with the body part or tangible object they used in the process. The technique can be deployed using minimal hardware, as demonstrated by *MatchPoint*, a generic computer vision-based implementation of the technique that requires only a webcam. We explore the design space of spontaneous spatial coupling, demonstrate the versatility of the technique with application examples, and evaluate MatchPoint performance using a multi-directional pointing task.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords

User input; Input techniques; Motion-matching; Pointing; Gesture input; Touchless input; Bodily interaction; Vison-based interfaces; Computer vision

INTRODUCTION

With the evolution of technology and computer vision, touchless bodily interaction has been brought into the main stream. Using the body as an input device frees the user from having to carry or use different physical devices and remote controls, a significant trait in a world where computing devices are ubiquitous and embedded in our everyday life.

In this work we introduce *spontaneous spatial coupling*, a hybrid technique of motion-matching and pointing that allows users to temporarily acquire a pointer, as illustrated in Figure 1. Controls are presented to the user as moving targets that are differentiable by their movement. To activate a control the user matches its motion using any movement they can generate (e.g. a body part or an object they move). Upon synchronisation with a displayed target, a spatial coupling is created between the user's input modality and the control. The spatial coupling is temporary for the purpose of a particular interaction.

Pointers can be acquired on-demand, using any type of movement captured by the input device, to be used for manipulation of individual controls or entire interfaces. Users can decouple from a pointer whenever they choose, providing the flexibility to change input modality in the case of fatigue, or for situational or contextual reasons. Everyday objects can be coupled to controls and left in place for prolonged periods, providing the opportunity to create unique tangible user interfaces. Multiple pointers can be instantiated, to support single users

for bi-manual, multi-modal and multi-point interaction, and multiple users for simultaneous engagement and collaboration.

MatchPoint is a computer vision-based implementation of spontaneous spatial coupling. Matchpoint is highly deployable as it requires only a webcam as minimal hardware, and enables users to interact flexibly with minimal constraints on their pose and ways of providing input. The system is based on display of controls with orbiting targets, and accepts any form of movement in its field of view as matching input. MatchPoint can detect and track multiple pointing instances in parallel. Due to its generic approach to motion detection, the system does not require any calibration or training. As the system requires only a webcam, it can be deployed in many application domains, including on large displays, tablets, laptops, and smartphones.

Our aim in this work is to define spontaneous spatial coupling as a new interaction technique and to explore the opportunities it affords. We advance theoretical and practical understanding of the technique through the following contributions:

- Definition of the properties that define spontaneous spatial coupling, and an exploration of the design space.
- MatchPoint, a webcam-based implementation of spontaneous spatial coupling which uses generic computer vision processing to accept any form of input.
- Practical examples, built for MatchPoint, which demonstrate the versatility of spontaneous spatial coupling for different interaction techniques.
- Evaluation of MatchPoint as an input device for pointing using the ISO 9241-9 multi-directional pointing task.

BACKGROUND AND RELATED WORK

Pointing is a fundamental interaction principle that draws on human spatial abilities and skills. The principle is at the core of graphical user interfaces on our desktops, tablets and smartphones, but naturally extends to touchless interaction with displays that are remote, shared, large, public, ambient or used in settings that prohibit touch for hygienic reasons [9, 21, 7, 61, 55, 46]. A range of devices exist for remote pointing but our interest is in computer vision methods that support users in pointer control without the need for mediating devices. Related work generally assumes use of the hands for pointing (e.g., [61, 52, 5, 34, 13]) but work in other areas has shown that humans are equally natural at pointing with other parts of their body (literally, from head [45, 38] to toe [58]). We reflect this in an approach that is input-agnostic and supports any body movement to be adopted for pointing, contrasting existing systems that are optimised for specific modalities such as tracking of hand gestures [48], head pose [54], or feet [56].

Conventional user interfaces support pointing by tightly coupling a pointing device with a display surface, or by integrating pointing and display. Mapping the user's movement to the display is a challenge for touchless pointing, as the user's input space is not straightforward to discover. Ill-conceived mapping may exacerbate fatigue issues, commonly referred to as "Gorilla Arm" when using the hands [29], or require the user to be centred to the input device. Jude *et al.* showed that accuracy can be maintained when pointing on small displays if the user is able define their input space during a calibration phase [34]. As we will show, spontaneous spatial coupling can avoid explicit calibration of the user's input space, instead using the user's range of movement during the motion-matching phase to define the mapping between the modality and display. Touchless pointing often suffers from the "Midas Touch" problem when the user has no means to 'turn-off' the pointer, as it is not always evident if a gesture is directed towards the display [39]. Our technique reduces the problem, as pointers are instantiated temporarily only when the user signals their intent to interact by synchronising with a displayed motion.

Motion-matching is an alternative selection mechanism to pointing, relying on the ability of users to couple with motion displayed at the interface [57]. First explored by Williamson and Murray-Smith [63], motion-matching has been used with a variety of different input modalities, including the mouse [63, 24], eye gaze [47, 59, 22], and recently touchless interaction [13, 17, 16]. PathSync demonstrated the discoverability, intuitiveness and multi-user capacity of motion-matching for hand-based gestures [13], while TraceMatch showed users' capacity to synchronise using different input modalities [16]. TraceMatch also introduced a webcam-based implementation of motion-matching that accepts any form of movement as input [17], an approach we adopt for the motion-matching phase in MatchPoint. Prior work proposed motion-matching as an alternative to spatial coupling, whereas we combine the two principles to leverage their respective advantages.

Pointing is usually based on a rigid coupling of a pointing mechanism with a display. Our concept differs fundamentally in supporting temporary creation of pointers, on demand for the purpose of a spontaneous interaction. Prior work has demonstrated spontaneous coupling of smartphones for pointing on public displays [4, 10], contrasting our work where users do not require any device. In other work, pointers have been dynamically mapped based on context, for instance of the user's gaze attention to different displays [19], whereas our approach provides users with explicit control over where and when to create a pointer.

Spontaneous spatial coupling as implemented in MatchPoint extends to dynamic coupling of tangible objects with on-screen functionality. When the user moves an object in synchrony with a displayed control, the object becomes a pointing device. In this sense, our work relates to graspable interfaces that use physical tokens for pointing [25], token-based interfaces that employ real-world props as handles for spatial controls [30], and tangible user interfaces that support ad hoc coupling of physical inputs [27, 60, 8]. MatchPoint supports such forms of physical control in a highly dynamic manner, but moreover also allows for incidental use of objects, for instance when the user synchronises with a control while they happen to hold an object. Other work enabling tangible interactions around devices has relied on computer vision for object classification [2, 37, 65], whereas our technique enables spatial coupling and tracking of objects without the need for the system to have prior knowledge of the objects.

Pointing has mostly been studied with applications that involve sustained interaction. In contrast, spontaneous spatial

coupling is geared toward use in contexts where users interact on impulse, where interactions are short, or where interactions occur on the side of other activity. Such forms of interaction are becoming more typical as we engage with increasing numbers of devices in our environments, highlighting a need for users to have instant control "right here, right now" [40]. Spontaneous spatial coupling addresses this need with a low-effort method for instant, yet expressive control.

SPONTANEOUS SPATIAL COUPLING

The interaction principle behind spontaneous spatial coupling is defined by five properties:

1. Distinct motions displayed to the user represent controls available for interaction;
2. A user's intent to interact with a control is expressed through movement corresponding to the control's motion;
3. The selection of a control is determined by the correlation between the system's output and the user's input;
4. Upon selection a spatial coupling is created between the user and the underlying functionality of the control;
5. The user is able to decouple from the control at will.

The interaction involves a phase of motion-matching followed by pointer control. The matching phase can be based on any shape of motion, and method for determining a correlation. Considerations for the design include how distinctive the motion is (to accidental matches), how easily and efficiently users can match it, and how reliably and robustly it can be detected. Velloso *et al.* provide a comprehensive review of design considerations for motion correlation interfaces [57].

Any input modality that produces motion can be used for both the matching phase and subsequent input. The matching phase results in a specific spatial coupling between the user and the control's underlying functionality. The coupling can be interpreted as a pointing device for which a cursor is instantiated, or as a device for describing gestural input. The output device needs to be able to present motion to facilitate the coupling, but once the user is coupled other types of feedback can be used for interaction (e.g. audio feedback when controlling the volume of a radio).

In the following, we discuss system design considerations to take into account when designing controls for spontaneous spatial coupling.

Control-Display Gain

The CD gain of the pointer can be set according to the size of the user's movement in the motion-matching phase. This approach assumes the user's movement range during the motion-matching phase is indicative of the movement range used when spatially coupled with a control. This may not always hold true and one might want to define the CD gain to increase the allowed range of movements, for example when manipulating objects on very-large displays.

Transfer Function

Absolute control maps provide a fixed gain so that the user's movement is directly mapped to the on-screen controls. To allow for greater precision, relative control maps, such as pointer-acceleration-based transfer functions, can be used in conjunction with techniques such as semantic pointing [7]. A drawback of relative control maps is the need for the input device to provide the ability to clutch, temporarily disabling the gesture, in order to allow the user to reposition themselves.

Pointer Starting Position

The user is in motion at the point of synchronisation with a moving target. If the system switched immediately into pointing mode, the pointer would move in continuation of the motion described for matching. For tasks such as parameter control or spatial selection, the user may wish to start the interaction from a well-defined position (e.g. the option currently selected). To allow this, the system can indicate when the match is detected, wait for the user to stop their matching motion, and enable pointing only after the matched input modality has become stationary. This may not always be required, for example when the pointer is used to represent an on-screen cursor using an absolute control mapping.

Pointer Termination

A range of mechanisms are possible for decoupling from a control. A pointer could terminate after completion of a single task for which it was instantiated, or after a pre-set time of inactivity. It is also possible to have the user explicitly trigger pointer termination, for instance using dwell or goal-crossing techniques [1]. If only one of the axes is used for input, users can use the other to signal task completion. If the coupling is used for gestural input, a specific gesture can be included for decoupling. Other than such generic techniques, specific implementations might afford device dependent decoupling mechanisms (e.g. using the depth axis of a depth sensor).

System Visibility

Moving targets are displayed to the user for acquisition and selection. The dynamic nature of the controls may be visibly distracting if the user is focussing on the display and has no intention of interacting with the system for prolonged periods of time. To overcome this the moving targets can be hidden from the user by assigning a specific control to hide the targets, or after a period of time with no input from the user. To display the moving targets, generic gestures can be used, e.g. moving an input modality in a full circle. There may be no need to hide the moving targets for applications where the user's main focus is not on the display.

MATCHPOINT

MatchPoint is a webcam-based implementation of spontaneous spatial coupling that accepts any type of movement as input. The system consists of two main processing pipelines: the motion matching pipeline which allows the user to select a control and acquire a pointer; and the tracking pipeline which allows the user to manipulate the pointer by providing a spatial coupling between the user's input modality and the control. Instances of the tracking pipeline can run in parallel with the motion-matching pipeline to allow a single user to acquire multiple pointers, or for multiple users to acquire a pointer.

Figure 2. Initialisation of the MatchPoint tracker once a user is in synchrony with an Orbit. Left: One matched feature point showing its trajectory (green) with fitted circle (red and blue). Centre: The result after connected-component labelling of candidate pixels that matched the motion of the feature point (green), calculated using dense optical flow. Right: The region of interest of the object to be tracked (green).

Motion-Matching

For motion-matching we use the TraceMatch processing pipeline, introduced by Clarke *et al.* [17]. TraceMatch uses *Orbits*, introduced by Esteves *et al.* [22], as input controls which consist of an orbiting target around a circular widget, a motion that is not likely to be reproduced accidentally by the user. FAST feature points [49] are detected and tracked using the pyramidal Lucas-Kanade optical flow algorithm [43, 11]. Each feature point that exhibits a minimum amount of movement is compared to all of the Orbis currently displayed to the user. A match is confirmed if the minimum Pearson correlation coefficient between either the x or y axis of a feature point and an Orbit is above a minimum threshold, and if the feature's trajectory can be successfully fitted to a circle with the appropriate arc length using RANSAC.

Spatial Coupling

To provide the spatial coupling between the user and the control we first initialise a region of interest (ROI) in the frame relating to the input modality, before tracking it in subsequent frames using a modified version of the Median Flow tracker [35]. We further process the output to remove pointer jitter introduced by image noise.

Tracker Initialisation

The output of the matching process is one or more feature points, however we ideally want to track as much of the body part, or object, which activated the control in order to improve downstream tracking performance. To do this we calculate the dense optical flow of the scene and compare the matched feature points' trajectories with the dense optical flow information, see Figure 2. First, we calculate the trajectories for all n matched feature points, $A = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$, between frames t and $t-1$ using their respective x and y coordinates, where the match occurred at frame t:

$$\mathbf{a}_i = \left(x_i^{t-1} - x_i^t, y_i^{t-1} - y_i^t\right) \text{ for } i = 1, \ldots, n \tag{1}$$

We then calculate the Euclidean distance of each trajectory, $D = \{\|\mathbf{a}_1\|, \ldots, \|\mathbf{a}_n\|\}$, and the average angle of the trajectories, $\bar{\theta}$, using the trajectories' average unit vector, $\bar{\mathbf{a}}$:

$$\bar{\mathbf{a}} = \frac{1}{n}\sum_{i=1}^{n} \hat{\mathbf{a}}_i \tag{2}$$

Where $\bar{\theta} = \angle\bar{\mathbf{a}}$. Using the Farnebäck algorithm [23] we calculate the dense optical flow of the scene, $F = \{\mathbf{f}_1, \ldots, \mathbf{f}_N\}$, between frames t and $t-1$ for all N pixels, $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ where:

$$\mathbf{p}_i^t = (x_i, y_i) \text{ for } i = 1, \ldots, N \tag{3}$$

$$\mathbf{p}_i^{t-1} = (x_i + \Delta x_i, y_i + \Delta y_i) \text{ for } i = 1, \ldots, N \tag{4}$$

$$\mathbf{f}_i = (\Delta x_i, \Delta y_i) \text{ for } i = 1, \ldots, N \tag{5}$$

We then identify pixels that matched the motion of the matched feature point(s) by assessing whether the ith pixel satisfies the following equations:

$$(1 - th_d)\min\{D\} < \|\mathbf{f}_i\| < (1 + th_d)\max\{D\} \tag{6}$$

$$\bar{\theta} - th_\theta < \angle\mathbf{f}_i < \bar{\theta} + th_\theta \tag{7}$$

Where $\|\mathbf{f}_i\|$ is the magnitude of the dense flow at pixel i, $\angle\mathbf{f}_i$ is the angle of the dense flow at pixel i, th_d is a distance threshold between 0 and 1, and th_θ is an angle threshold measured in radians. For this implementation we use values of 0.25 and $\pi/8$ for th_d and th_θ respectively. This produces a mask of candidate pixels which is further processed to remove candidates resulting from image noise or background movement.

Finally, we use connected-component labelling to find candidate groups from the candidate pixels. We seed the connected-component labelling with $10px \times 10px$ areas around the matched feature points because errors can be introduced in the tracking of individual feature points during the motion-matching phase, i.e. a feature point is offset from the body part/object. We then fit a rotated rectangle around all the candidate groups connected to the matched feature point(s), resulting in the initial ROI for the tracker which will be tracked in subsequent frames. In the event there are no candidate groups, we initialise a minimum rectangular ROI $(30px \times 30px)$ around the matched feature points.

Control-Display Gain

During the initialisation phase we also calculate the control-display (CD) gain. The output of the matching process returns a fitted circle for each matched feature point which indicates the ideal trajectory of the user's movement when matching the control (i.e. without any noise). We use this as an indication of the range of movement for which the user is comfortable using, as it takes into account the input modality used and distance from the camera, e.g. a head movement may result in a much smaller radius than a hand movement. The CD gain, CD_{gain}, is calculated by taking the reciprocal of the average

radii, r, of the fitted circles from the matched feature points:

$$CD_{gain} = \left(\sum_{i=1}^{N} r^i / N \right)^{-1} \quad (8)$$

The CD gain is then multiplied by the appropriate distance depending on the context, e.g. for a screen cursor this would be the width or height of the screen, whereas for a single control this would be the width or height of the control.

Tracking

The body part or object to be tracked may not exhibit smooth movements. The tracking should cope with unpredictable movements and changes in perspective of the body part/object relative to the camera. We experimented using different trackers, including Median Flow [35], KCF [18], MIL [3], TLD [36], and OLB [26]. Preliminary testing indicated that non were suitable for this application so we instead use a modified version of Median Flow to track a rotated rectangle.

The first modification we made was the selection of points to track at each iteration using Median Flow. Kalal *et al.* recommend using a grid of equidistant points or the use of a feature detector, such as FAST [35]. However, the body part or object that we wish to track may not fill the whole of the ROI, so instead we use a grid spacing based on central polygonal numbers (aka. the Lazy Caterer's sequence) to reduce the chance of the tracker getting stuck on background objects.

The second modification was to introduce a "recalibration" phase for the tracker, which accounts for changes in the size and perspective of the body part/object that is being tracked, whilst ensuring that the ROI covers as little as the background as possible. For recalibration we follow the same steps for initialising the ROI, however instead of using the matched feature point's trajectory in Eq. 1 and 2 we use the trajectory of the ROI calculated using its centre, and instead of the matched feature points acting as the seeds to the connected-component labelling we use the centre of the ROI. As we have knowledge of the previous ROI we focus the search to a rectangular area around the centre of the ROI prior to calibration with width, $2 \times W$, and height, $2 \times H$, where W and H are the width and height of the ROI prior to recalibration. We remove the scaling of the ROI provided by the original Median Flow algorithm as this is performed during the recalibration phase.

The recalibration phase relies on the movement of the ROI, therefore we only perform the tracker recalibration when the magnitude of the ROI's trajectory is above a threshold, th_{ROI}, and at most every $t_{recalib}$ seconds to limit processing time. For our implementation we set th_{ROI} to 2 pixels, and $t_{recalib}$ to 500ms. The centre of the ROI is used as the reference point of the tracker (i.e. the point which updates the on-screen control). When we recalibrate the ROI, the centre may change which would cause a jump in the cursor from the perspective of the user. To avoid this we record the offset caused by the recalibration of the ROI and apply this to the output in subsequent frames so that the recalibration is unnoticeable to the user.

Reducing Pointer Jitter

Cameras are subject to image noise that affect tracker performance and result in unwanted movement of the pointer at the interface. To reduce the effects of noise we take the average position of the ROI using a dynamic moving window when the Euclidean distance between the centre of the tracker from the previous frame to the current frame, d_t, is less than d_{MIN} pixels. The size of the moving window, N_B is calculated as:

$$N_B = N_{MAX} - \lfloor \frac{d_t \times N_{MAX}}{d_{MIN}} \rfloor \quad (9)$$

Where N_{MAX} is the maximum size of the moving window in frames, MatchPoint uses 10 frames. The moving window introduces input lag, therefore the value of N_{MAX} should be carefully considered based upon the frame rate of the camera. The value of d_{MIN} will vary depending on the quality of the camera, for our implementation this is set to 2.5 pixels using a Logitech C920 webcam.

INTERACTION TECHNIQUES & APPLICATIONS

In the following we explore interactions that can be supported with spontaneous spatial coupling. We consider five cases:

- Single pointer $\rightarrow$ Multiple functionality – one pointer provides all the functionality in the interface;
- Multiple pointers $\rightarrow$ Multiple functionality – each pointer provides a different functionality;
- Multiple pointers $\rightarrow$ Single functionality – multiple pointers provide the same functionality;
- Parallel pointers – multiple pointers used in parallel;
- Tangible interfaces – creating temporary tangible interfaces using everyday objects.

The first case is the conventional case of "pointing as we know it". For all other cases, we present novel techniques and application demonstrators implemented with MatchPoint. All of the examples are applicable to both single and multiple users, as controls can be matched and spatially coupled in a non-exclusive manner.

Single Pointer $\rightarrow$ Multiple Functionality

Conventionally, touchless pointers use one pointer to control many on-screen input controls. This technique can be used with spontaneous spatial coupling, allowing users to decide which input modality to use, and in the event of fatigue to desist pointing and resume with another input modality. If only one pointer is required, it is also possible to replace the motion-matching stage with a generic motion gesture, as there is no need to differentiate between different controls.

Multiple Pointers $\rightarrow$ Multiple Functionality

Multiple pointers can be used to provide different functionality. Users can acquire different pointers depending on the interaction to be performed, removing the need for a user to navigate through an interface using a single pointer. This also allows the CD gain to be mapped according to the size of a control, as opposed to size of the display. To demonstrate this concept, we developed two applications for different scenarios: a TV remote control for the living room, and a video player for use when engaged in other physical activity (e.g., in the kitchen).

Figure 3. TV remote control prototype, showing Orbits for: channel up and down (left), TV guide and channel select (middle), and mute toggle and volume control (right).

TV Remote Control

Conventionally, a TV remote control is shared, and it must be passed from person to person. Gesture control for TV has been investigated in prior research, driven by users' desire to have instant control [40], but this has primarily focussed on library-based gestural techniques [14, 31, 33]. The level of interaction with TVs is increasing with the integration of "smart" features, while users primarily focus on the content displayed, or may watch it in the background whilst performing other tasks with minimal interaction. This provides an exemplar application space for MatchPoint.

The TV remote control application features controls for changing the channel up and down, muting the volume, changing the volume, selecting a channel from a list, and showing a TV guide (Fig. 3). Changing the channel up and down, and muting the volume, are binary choices and do not require spatial coupling, therefore these are implemented as motion-matching only controls.

Upon selecting the volume control, a one-directional slider is presented to the user (Fig. 4, left). The control waits until the user's input modality is stationary prior to displaying the control, allowing the user to change the volume relative to the current volume of the TV. Movement in the y-direction sets the volume level, and movement in the x-direction either cancels the control (when user moves left), or confirms the new volume level (when user moves right).

The channel selection control displays a one-directional carousel control (Fig. 4, right). This also waits for the user's input to be stationary prior to activating to ensure the user starts searching the channels from their currently selected channel. Movement in the x-direction scrolls through the channels,

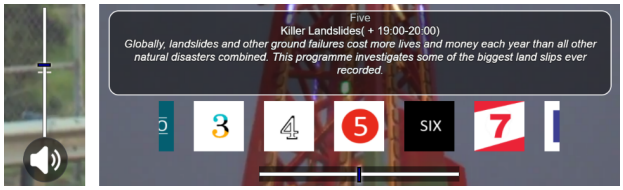

Figure 4. Volume slider which can be controlled with movements in the y-axis (left), and channel selection control showing the programme details and slider to indicate the position of the user's movements in the x-axis which controls the carousel (right).

Figure 5. TV remote control prototype, showing the TV guide.

dwelling on a channel displays the programme details, and movement in the y-direction either cancels the control (down) or changes the channel to the current selection (up).

For the TV guide the user is presented with a cursor for navigation (Fig. 5). Dwelling on a programme displays the details in the upper-most box, which can be selected to present the user with a number of options depending on the programme, including watch now, set a reminder, start a recording, or close the TV guide. The user can also close the guide by dwelling at the edge of the interface.

Video Player

Building upon the TV context we developed a video player application. Video guides for practical tasks, such as cooking or car maintenance, are popular on platforms such as YouTube. Users will often watch the videos in-situ on a portable device, such as a laptop, tablet or mobile phone, whilst performing a task. Interactions with the video guide will be relatively sparse, such as pausing and rewinding, as the user is primarily focussed on performing the actions demonstrated in the video guide. Spontaneous spatial coupling allows the user to perform other tasks whilst interacting with the video player, e.g. using cooking utensils in the kitchen whilst cooking.

The video player features controls that allows the user to play or pause, change the playback, navigate to a specific time, and mute or change the volume (Fig. 6). The controls for play/pause and muting the volume are binary choices, therefore we use motion-matching only controls.

Figure 6. Video control prototype showing Orbits for: play/pause (left), video progress bar and playback (middle), and volume control (right).

Figure 7. Playback control for the video control prototype where movements in the x-direction select different playback options.

The video playback control allows the user to trigger playback of the video through movement in the x-direction, while movement in the y-direction exits the control (Fig. 7). Upon activation the control waits for the user's input modality to become stationary, ensuring that the play button is selected when the interaction starts.

The control allowing users to navigate to a specific time in the video uses movement in the x-direction to select the time, and movement in the y-direction to either confirm the selection (by moving up) or cancel the control (by moving down). Upon activation the user must pause their motion briefly so that search can resume from the current time in the video.

Multiple Pointers → Single Functionality

In the above examples multiple users can interact with the controls, but only one can interact with a specific control at any given time (e.g. two people can't change the volume at the same time). In some instances, it may be desirable for multiple users to acquire a pointer with the same functionality, for example to contextualize their discussions [46].

Whiteboard Pointing Prototype

A scenario in which this would be beneficial is a meeting room where users are remote from the display but would like to indicate a position on the screen. For this we designed a "whiteboard" pointing prototype featuring a control to allow users to acquire a cursor (Fig. 8). Upon selection, users acquire a cursor with a unique colour to allow multiple people to control a pointer and provide input to a shared space as and when required.

Parallel Pointers

Users can acquire multiple pointers at the same time. The functionality of the pointers used in parallel may be:

- Unrelated – control of one pointer does not affect the other(s)

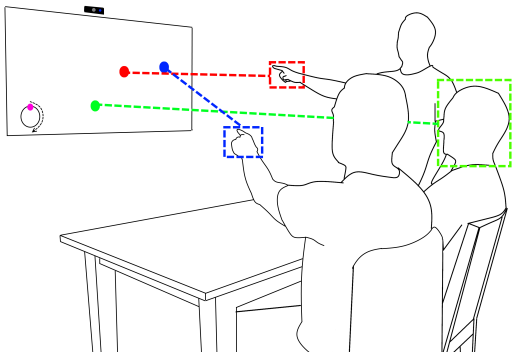

Figure 8. White board pointing prototype demonstrating multiple cursors (green, red, and blue). Dotted lines and rectangles represent the input modality and user controlling the cursor. The Orbit (pink) shows the colour of the next cursor to be generated.

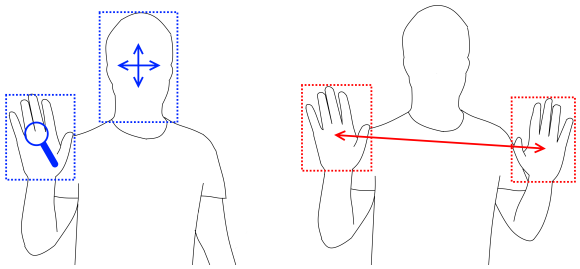

Figure 9. Two prototypes for parallel pointing. Multi-modal pan and zoom (left): one input modality controls the pan (the head), the other controls the zoom (the hand). Bi-manual pointing (right): the centre point between the hands determines the pan position, the distance determines the zoom, and the angle determines the rotation.

- Loosely coupled – the pointers affect the same object, but can be used individually
- Tightly coupled – interaction results from the relationship between the pointers

For unrelated and loosely coupled pointers, parallel usage allows users to complete tasks in less time, or to manipulate one control based on the state of another. Tightly coupled pointers offer more complex interactions by using the pointers' spatial relationship with each other, however they must be used at the same time and can not be used individually.

Object Manipulation Prototype

To demonstrate loosely and tightly coupled parallel pointing, we created two prototypes for object manipulation (Fig. 9). The first is a multi-modal prototype designed to be used with any input modality. It consists of two loosely coupled input controls: one to pan the object, and the other to zoom. The pan control uses the x and y directions of the user's first input modality to position the object. The zoom control uses movement in the y-direction of the user's second input modality to zoom in or out, using an acceleration-based transfer function to control the level of zoom. Movement in the x-direction is used to enable clutching (by moving to the left) or exits the control (by moving to the right).

The second prototype demonstrates bi-manual input, designed specifically for the hands using two tightly-coupled controls. Object manipulation is supported in a way that is familiar from touch-screens: the distance between the hands determines the zoom, the centre point between the hands determines the pan position, and the positions of the hands relative to each other determines the rotation.

The bi-manual prototype requires both controls to be activated in order to enable the parallel pointing interaction. However, it provides the user with three functions (pan, zoom and rotate) using only two pointers, contrasting the multi-modal approach which only provides two functions (pan and zoom). Note that the system does not identify body parts, and that it would be possible for users to acquire the controls with other than the intended modalities. To avoid this, an interface should convey to the user if a specific input modality is required.

Tangible Interfaces

Other than creating a coupling with a part of their body, users can also move a tangible object in synchrony with a displayed

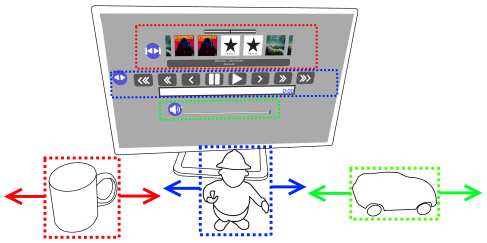

Figure 10. A tangible interface created with MatchPoint. The cup controls the playlist, the toy figure controls the playback, and the toy car controls the volume. Moving the objects left and right changes the value of the respective control.

motion. The result is a spatial coupling between tangible object and control. The use of objects in this way presents a distinct case, as they become tangible intermediaries between user and control. This has interesting affordances, as a user can leave an object stationary in between interactions, with a persistent coupling. Objects can also afford specific manipulations depending on their shape and weight, for example nudging, rolling and tilting.

Graspable Music Player

To demonstrate this concept we developed a tangible music player interface (Fig. 10). Three controls are displayed to the user allowing them to change the playlist, volume, or the playback. Once the user couples a physical object to a control the system waits for the user to position the object into its starting position. The control is not fully activated until the object remains stationary for an extended period of time (e.g. 4 seconds). All input controls in this example utilise movement in the x-axis to manipulate the respective control, e.g. moving the toy car left lowers the volume, moving it right increases the volume. The objects remain paired with the controls until they are removed from the camera's field of view.

MULTI-DIRECTIONAL POINTING TASK EVALUATION

We conducted a two-directional Fitt's Law study, based on the ISO 9241-9 standard, to understand the performance of MatchPoint as an input device for pointing and to provide insights into pointing performance with different input modalities. We compare three input modalities (head, hand and cup-in-hand) to investigate their throughput and other pointing characteristics in a simulated living room environment. We chose a simulated living room environment to test how the system performed when the user was at a larger distance (>2m) from the input device.

Task

Ten circular targets of diameter W were displayed in a circular configuration with a radius of $A/2$. In order to avoid possible confounds we used Guiard's Form x Scale design [28], with three levels of W (50px, 100px, 200px) and one level of A (900px), resulting in three unique index of difficulty (ID) values of 4.24, 3.32, and 2.46 respectively. When calculating the throughput we use the effective index of difficulty, ID_e, by measuring the effective values of A and W, which take into consideration the speed/accuracy trade-offs participants make when completing the task:

$$ID_e = \log_2(A_e/W_e + 1) \tag{10}$$

Where A_e is the average movement distance observed [53], and W_e is the standard deviation of endpoints, defined as:

$$W_e = 4.133 \times SD_{x,y} \tag{11}$$

Where, $SD_{x,y}$ is the bivariate endpoint deviation, defined as [64]:

$$SD_{x,y} = \sqrt{\frac{\sum_{i=1}^{N}\left(\sqrt{(x_i-\bar{x})^2+(y_i-\bar{y})^2}\right)^2}{N-1}} \tag{12}$$

Dwell was used as the selection process, with a dwell time of 240ms to simulate the time taken to press a button [50]. We also measure the number of target re-entries, as defined by MacKenzie *et al.* [44].

Participants and Apparatus

We recruited 12 participants to undertake the study (mean = 28.1, SD = 3.9). Six participants were female, and one was left-handed. None of the participants had used the MatchPoint system prior to the study.

The study was conducted in a simulated living room environment, using a Samsung 55" Smart TV (1920 x 1080) as the display. An unmodified off-the shelf webcam was used as the input device to the MatchPoint system and captured a 640 x 480 region of interest from a 1920 x 1080 frame. The region of interest was used to ensure that only movement relating to the study was captured by the webcam. Participants were seated on a couch 2.23m from the TV (based on a TV size to viewing distance calculator). For the cup-in-hand input modality participants were asked to hold a cup half-filled with water to simulate holding a drink.

Procedure

At the beginning of the study participants completed a demographics questionnaire and were introduced to the MatchPoint system. They were instructed to relax and work comfortably whilst performing the tasks as quickly and as accurately as possible. No instructions were given regarding how to hold the cup or position the hand when matching and pointing.

For each input modality, participants undertook three sets (one for each ID), where a set consisted of five blocks of 10 target selections. The first block was used as a warm-up, with the remaining four being used for data collection. At the start of each block the users acquired a cursor by matching the motion of an Orbit with the specified input modality. Participants were instructed to let the researcher know if they required a break in between blocks due to fatigue. A Latin square design was used to counterbalance for input modality, and the order in which the IDs were presented and the starting position of the first target were randomised. The movement time was only measured after the first target was selected. The study ended with a brief verbal discussion to gain feedback on the system.

Excluding the warm-up blocks the study involved 12 participants $\times$ 3 input modalities $\times$ 3 IDs $\times$ 4 blocks $\times$ 10 trials per block = 4320 trials.

Results

Movement times, shown in Figure 11, were analysed using a one-way repeated measures ANOVA to determine whether any statistically significant differences existed for different input modalities. The data was normally distributed, as assessed by boxplot and Shapiro-Wilk test, $p > .05$, and the assumption of sphericity was not violated, as assessed by Mauchly's test of sphericity, $\chi^2(2) = 3.903, p = .142$. The test revealed a significant difference between movement times ($F_{2,22} = 73.166, p < .001$). Post-hoc pairwise comparisons with Bonferroni corrections revealed the movement time of the head (3.12s) was significantly higher compared to both the hand (2.37s) and cup (2.54s), at $p < .001$. There was also a significant difference between the movement time of the hand and cup, at $p = .035$.

The grand throughputs for each input modality were calculated using the mean of means approach [53]. A one-way repeated measures ANOVA was used to determine whether there were statistically significant differences between throughputs for different input modalities. The data was normally distributed, as assessed by boxplot and Shapiro-Wilk test, $p > .05$, and the assumption of sphericity was not violated, as assessed by Mauchly's test of sphericity, $\chi^2(2) = 1.856, p = .395$. A significant difference between input modalities was revealed ($F_{2,22} = 54.617, p < .001$). Post-hoc pairwise comparisons with Bonferroni corrections showed that the throughput of the head (1.03 bits/s) was significantly lower compared with both the hand (1.35 bits/s) and cup (1.25 bits/s), at $p < .001$. There was not a statistically significant difference between the throughputs of the hand and the cup, at $p = .059$. Using linear regression, we then developed Fitts' Law models of the form:

$$MT_{modality} = a + b \cdot ID_e \tag{13}$$

Where, $MT_{modality}$ is the predicted movement time for an input modality, measured in seconds. The parameters for a and b, and the R-squared model fit values are given in Table 1.

We were interested to see whether a statistically significant difference existed between target re-entries for different input modalities and target sizes (Fig. 12). For this, we used a two-way repeated measures ANOVA. The data was normally distributed, as assessed by boxplot and Shapiro-Wilk test ($p > .05$), and in all cases the assumption of sphericity was not violated, as assessed by Mauchly's test of sphericity ($p > .05$). We discovered significant main effects for input modality ($F_{2,22} = 24.836, p < .001$), size ($F_{2,22} = 44.341, p < .001$),

Input modality	*a*	*b*	R^2
Head	-1.000	1.345	0.766
Hand	0.417	0.624	0.556
Cup	-0.118	0.857	0.795

Table 1. Fitts' Law parameters and model fits for each input modality.

and a significant interaction for input modality × size, ($F_{4,44} = 11.472, p < .001$). We further analysed the main effects using Bonferroni corrected pairwise comparison of means.

For the main effect of input modality, we observed that the head had a significantly higher number of target re-entries (0.541) compared with both the hand (0.299) and cup (0.304), $p < .001$. There was no statistical significant difference between the hand and the object, $p = 1.0$. The main effect for size showed that smaller targets (0.649) resulted in a higher number of target re-entries compared with medium targets (0.323), which in-turn resulted in a higher number of target re-entries compared with larger targets (0.172). In all cases the results were statistically significant at $p < .005$.

To analyse the input modality × size interaction, we performed a one-way repeated measures ANOVA for each size to assess which input modalities, if any, caused a significant difference to target re-entries. For all sizes, Mauchly's test of sphericity was not violated. There was no significant difference between input modalities for the large size, however there were significant differences for both the medium ($F_{2,22} = 5.513, p < .011$) and small sizes ($F_{2,22} = 22.546, p < .001$). Post-hoc Bonferroni corrected pairwise comparisons revealed the head had a significantly higher number of target re-entries compared with the cup for medium target sizes, at ($p = .011$), and both the hand and cup for small target sizes, at ($p < .005$). No false motion-matching activations occurred during the study.

Study Discussion

The larger throughput and number of target re-entries for the head may be a result of both user and system performance. The head is used much less frequently for everyday pointing tasks compared with the hand, and the smaller range of movement at a distance of over 2m from the camera equates to fractional changes per pixel between frames. The moving window used to reduce cursor jitter for very small movements introduces a time lag which could have affected the user's fine-grain pointing performance. This can be alleviated using augmented cursors, which have been shown to improve target acquisition of smaller targets for gestural interaction [20].

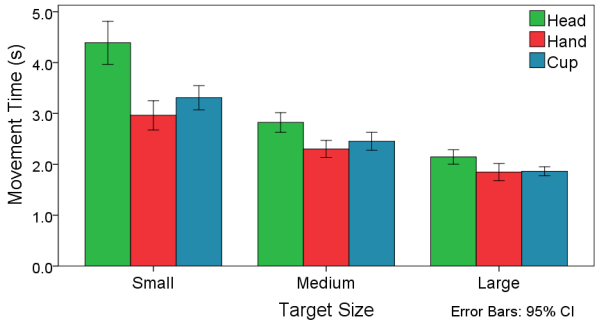

Figure 11. Movement times for each target size and input modality.

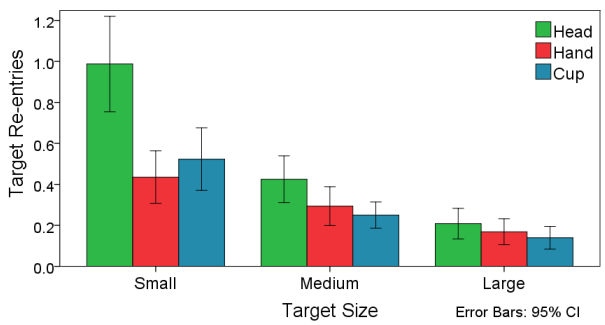

Figure 12. Target re-entries for each target size and input modality.

Distance to Device	Study	Input Modality	Input Device	TP (bits/s)
<1m	[12]	Hand	Leap Motion	2.8
	[15]	Hand	MS Kinect v1	1.42
	[32]	Head	Head-mounted marker	1.61
	[48]	Hand	MS Kinect v1	2.1
	[62]	Head	Optical IR markers	1.40
>1m	[6]	Head	MS Kinect v2	2.3
	[6]	Hand	MS Kinect v2	2.45
	[42]	Head	MS Kinect v1	0.75
	[51]	Hand	MS Kinect v1	1.19-1.38*
Unknown	[41]	Hand	Wii Remote	2.97
	[50]	Hand	MS Kinect v1	0.5-2.0**
	[51]	Hand	Swiss Ranger 4000	0.75-1.57*

Table 2. Touchless input devices used in previous studies that used the ISO 9241-9 multi-directional pointing task to assess throughput (TP) for hand and head gestures. * indicates a range of throughputs due to different selection techniques. ** indicates estimated throughput.

Table 2 details prior work which used the ISO 9241-9 multi-directional pointing task to examine the throughput of input devices for head and hand gestures. A direct comparison of devices cannot be made due to the variability of participants, distances to the input device, measurement of endpoint deviation, selection of IDs, and difference in selection techniques. However, it appears that MatchPoint has a similar throughput to the first version of the Microsoft Kinect when used at larger distances. It is also important to note that we used a dwell time of 240ms to simulate the time taken to press a button. This may be suitable for some tasks, but for others a larger dwell time may be needed to reduce false detections when a user hovers over a control.

DISCUSSION

Spontaneous spatial coupling can support wide-ranging applications by enabling flexible touchless input over a distance. At the core of the concept is the motion-matching phase – it empowers users to simultaneously select the function they wish to control, and the input modality to use (implicit in their action). The selections made by the user, and additional contextual information such as scale and range of motion observed in the matching process, in turn enable input to be uniquely tailored to the context. As shown, this encompasses tracking of the specific modality of choice as a pointing device, calibration of the control display gain based on context, and the possibility to map input in a task-specific manner to parameters of the selected function.

The dynamic appropriation of "anything the user can move" as a pointing device presents a new design opportunity, inviting exploration of mappings that might not be general purpose but fitting for specific contexts. As shown, our concept extends to spatial coupling of multiple controls at a time, by one or multiple users, with body parts and/or objects. This opens up a compelling design space, for which we have provided an initial framing and demonstrated a range of novel techniques.

MatchPoint provides a highly deployable implementation of spontaneous spatial coupling. The system requires only an off-the-shelf RGB camera, and uses low-cost computer vision techniques that are able to track input without the need for recognition of objects or body parts. Other sensing modalities could be considered for spatial coupling, for example depth sensors to extend motion-matching and pointing into 3D, or inertial measurement units, to leverage sensors that are widely deployed in mobile and wearable devices.

MatchPoint's ability to accept any form of input is compelling as it enables users to choose a form of input that is convenient in a given context. As shown, users perform well with MatchPoint for pointing over a distance, but the modality can affect performance – raising the question of when to design for flexible choice versus specific modality. Based on its ability to accept any form of input, MatchPoint could also be deployed as an accessibility device, to provide users who can not operate a conventional mouse with a flexible alternative.

The motion-matching phase in MatchPoint is based on circular motion, adopted for the purpose as it provides uniformity to acquisition of controls. However, the system could be extended to support matching with any shape of motion by using a generic model fitting approach. Matching against any type of motion could provide designers with additional opportunities, for example selection of graphical objects for manipulation by tracing their outline, or use of polygonal motion paths as corners could help users synchronise.

There are several limitations in the current implementation of MatchPoint. The tracker used for spatial coupling does not handle occlusion, and simultaneous motion could result in tracking errors when feature points are detected for body parts connected to the user's desired input modality (e.g. tracking of the elbow when using the hand). If multiple people perform the exact same motion at the same time the system might also attempt to track their combined movements. The system may also attempt to track the hand when it is removed from a physical object when creating tangible interfaces. These limitations could be overcome by incorporating object recognition and segmentation of body parts, which would also open up the possibility for designers to present different interfaces for a control depending on which input modality is being used.

CONCLUSION

Spontaneous spatial coupling is a powerful concept for touchless input as it empowers users to dynamically appropriate any part of their body, or object they hold, as a pointing device. The concept leverages motion-matching as an intuitive method for users to select a control while implicitly creating a spatial coupling that is tailored to the context, supporting the acquisition of a pointer as and when needed. The concept also opens up an entirely new design space for interactions that leverage spontaneous coupling of multiple controls at a time, by one or multiple users, with different body parts, or with objects as tangible intermediaries.

MatchPoint is a systems contribution that provides a complete implementation of spontaneous spatial coupling. The system lends itself to wide deployment as it only requires an off-the-shelf camera and computer vision for detection, matching and tracking of motion input. The system is able to take input of any form, and adapts the control display gain to provide users with a comfortable input range.

REFERENCES

1. Johnny Accot and Shumin Zhai. 2002. More Than Dotting the I's — Foundations for Crossing-based Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, USA, 73–80. DOI: http://dx.doi.org/10.1145/503376.503390

2. Daniel Avrahami, Jacob O. Wobbrock, and Shahram Izadi. 2011. Portico: Tangible Interaction on and Around a Tablet. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 347–356. DOI: http://dx.doi.org/10.1145/2047196.2047241

3. Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. 2009. Visual tracking with online multiple instance learning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 983–990.

4. Rafael Ballagas, Michael Rohs, and Jennifer G. Sheridan. 2005. Sweep and Point and Shoot: Phonecam-based Interactions for Large Public Displays. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, New York, NY, USA, 1200–1203. DOI: http://dx.doi.org/10.1145/1056808.1056876

5. Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. 2012. MultiPoint: Comparing Laser and Manual Pointing As Remote Input in Large Display Interactions. *Int. J. Hum.-Comput. Stud.* 70, 10 (Oct. 2012), 690–702. DOI: http://dx.doi.org/10.1016/j.ijhcs.2012.05.009

6. Ana M Bernardos, David Gómez, and José R Casar. 2016. A Comparison of Head Pose and Deictic Pointing Interaction Methods for Smart Environments. *International Journal of Human-Computer Interaction* 32, 4 (2016), 325–351.

7. Renaud Blanch, Yves Guiard, and Michel Beaudouin-Lafon. 2004. Semantic Pointing: Improving Target Acquisition with Control-display Ratio Adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 519–526. DOI: http://dx.doi.org/10.1145/985692.985758

8. Florian Block, Michael Haller, Hans Gellersen, Carl Gutwin, and Mark Billinghurst. 2008. VoodooSketch: Extending Interactive Surfaces with Adaptable Interface Palettes. In *Proceedings of the 2Nd International Conference on Tangible and Embedded Interaction (TEI '08)*. ACM, New York, NY, USA, 55–58. DOI: http://dx.doi.org/10.1145/1347390.1347404

9. Richard A. Bolt. 1980. "Put-that-there": Voice and Gesture at the Graphics Interface. *SIGGRAPH Comput. Graph.* 14, 3 (July 1980), 262–270. DOI: http://dx.doi.org/10.1145/965105.807503

10. Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. 2010. Touch Projector: Mobile Interaction Through Video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 2287–2296. DOI: http://dx.doi.org/10.1145/1753326.1753671

11. Jean-Yves Bouguet. 2000. Pyramidal implementation of the Lucas Kanade feature tracker. *Intel Corporation, Microprocessor Research Labs* (2000).

12. Michelle A Brown, Wolfgang Stuerzlinger, and others. 2014. The performance of un-instrumented in-air pointing. In *Proceedings of Graphics Interface 2014*. Canadian Information Processing Society, 59–66.

13. Marcus Carter, Eduardo Velloso, John Downs, Abigail Sellen, Kenton O'Hara, and Frank Vetere. 2016. PathSync: Multi-User Gestural Interaction with Touchless Rhythmic Path Mimicry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 13. DOI: http://dx.doi.org/10.1145/2858036.2858284

14. Ming-yu Chen, Lily Mummert, Padmanabhan Pillai, Alexander Hauptmann, and Rahul Sukthankar. 2010. Controlling Your TV with Gestures. In *Proceedings of the International Conference on Multimedia Information Retrieval (MIR '10)*. ACM, New York, NY, USA, 405–408. DOI: http://dx.doi.org/10.1145/1743384.1743453

15. Ngip-Khean Chuan and Ashok Sivaji. 2012. Combining eye gaze and hand tracking for pointer control in HCI: Developing a more robust and accurate interaction system for pointer positioning and clicking. In *Humanities, Science and Engineering (CHUSER), 2012 IEEE Colloquium on*. IEEE, 172–176.

16. Christopher Clarke, Alessio Bellino, Augusto Esteves, and Hans Gellersen. 2017. Remote Control by Body Movement in Synchrony with Orbiting Widgets: an Evaluation of TraceMatch. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 45 (Sept. 2017), 22 pages. DOI: http://dx.doi.org/10.1145/3130910

17. Christopher Clarke, Alessio Bellino, Augusto Esteves, Eduardo Velloso, and Hans Gellersen. 2016. TraceMatch: A Computer Vision Technique for User Input by Tracing of Animated Controls. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 298–303. DOI: http://dx.doi.org/10.1145/2971648.2971714

18. Martin Danelljan, Fahad Shahbaz Khan, Michael Felsberg, and Joost Van de Weijer. 2014. Adaptive color attributes for real-time visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1090–1097.

19. Connor Dickie, Jamie Hart, Roel Vertegaal, and Alex Eiser. 2006. LookPoint: An Evaluation of Eye Input for Hands-free Switching of Input Devices Between Multiple Computers. In *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments (OZCHI '06)*. ACM, New York, NY, USA, 119–126. DOI: http://dx.doi.org/10.1145/1228175.1228198

20. Ashley Dover, G. Michael Poor, Darren Guinness, and Alvin Jude. 2016. Improving Gestural Interaction With Augmented Cursors. In *Proceedings of the 2016 Symposium on Spatial User Interaction (SUI '16)*. ACM, New York, NY, USA, 135–138. DOI: http://dx.doi.org/10.1145/2983310.2985765

21. Scott Elrod, Richard Bruce, Rich Gold, David Goldberg, Frank Halasz, William Janssen, David Lee, Kim McCall, Elin Pedersen, Ken Pier, John Tang, and Brent Welch. 1992. Liveboard: A Large Interactive Display Supporting Group Meetings, Presentations, and Remote Collaboration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '92)*. ACM, New York, NY, USA, 599–607. DOI: http://dx.doi.org/10.1145/142750.143052

22. Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015. Orbits: Gaze Interaction for Smart Watches using Smooth Pursuit Eye Movements. In *Proc. of the 28th ACM Symposium on User Interface Software and Technology (UIST 2015)* (2015-11-01). DOI: http://dx.doi.org/10.1145/2807442.2807499

23. Gunnar Farnebäck. 2003. Two-frame Motion Estimation Based on Polynomial Expansion. In *Proceedings of the 13th Scandinavian Conference on Image Analysis (SCIA'03)*. Springer-Verlag, Berlin, Heidelberg, 363–370. http://dl.acm.org/citation.cfm?id=1763974.1764031

24. Jean-Daniel Fekete, Niklas Elmqvist, and Yves Guiard. 2009. Motion-pointing: Target Selection Using Elliptical Motions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 289–298. DOI: http://dx.doi.org/10.1145/1518701.1518748

25. George W. Fitzmaurice, Hiroshi Ishii, and William A. S. Buxton. 1995. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 442–449. DOI: http://dx.doi.org/10.1145/223904.223964

26. Helmut Grabner, Michael Grabner, and Horst Bischof. 2006. Real-time tracking via on-line boosting.. In *Bmvc*, Vol. 1. 6.

27. Saul Greenberg and Michael Boyle. 2002. Customizable Physical Interfaces for Interacting with Conventional Applications. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (UIST '02)*. ACM, New York, NY, USA, 31–40. DOI: http://dx.doi.org/10.1145/571985.571991

28. Yves Guiard. 2009. The Problem of Consistency in the Design of Fitts' Law Experiments: Consider Either Target Distance and Width or Movement Form and Scale. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1809–1818. DOI: http://dx.doi.org/10.1145/1518701.1518980

29. Darren Guinness, Alvin Jude, G. Michael Poor, and Ashley Dover. 2015. Models for Rested Touchless Gestural Interaction. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction (SUI '15)*. ACM, New York, NY, USA, 34–43. DOI: http://dx.doi.org/10.1145/2788940.2788948

30. Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. 1994. Passive Real-world Interface Props for Neurosurgical Visualization. In *Conference Companion on Human Factors in Computing Systems (CHI '94)*. ACM, New York, NY, USA, 232–. DOI: http://dx.doi.org/10.1145/259963.260443

31. Inwook Hwang, Hyun-Cheol Kim, Jihun Cha, Chunghyun Ahn, Karam Kim, and Jong-Il Park. 2015. A gesture based tv control interface for visually impaired: Initial design and user study. In *Frontiers of Computer Vision (FCV), 2015 21st Korea-Japan Joint Workshop on*. IEEE, 1–5.

32. Rados Javanovic and Ian MacKenzie. 2010. MarkerMouse: mouse cursor control using a head-mounted marker. *Computers Helping People with Special Needs* (2010), 49–56.

33. Soonmook Jeong, Jungdong Jin, Taehoun Song, Keyho Kwon, and Jae Wook Jeon. 2012. Single-camera dedicated television control system using gesture drawing. *IEEE Transactions on Consumer Electronics* 58, 4 (2012), 1129–1137.

34. Alvin Jude, G. Michael Poor, and Darren Guinness. 2014. Personal Space: User Defined Gesture Space for GUI Interaction. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 1615–1620. DOI: http://dx.doi.org/10.1145/2559206.2581242

35. Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. 2010. Forward-backward error: Automatic detection of tracking failures. In *Pattern recognition (ICPR), 2010 20th international conference on*. IEEE, 2756–2759.

36. Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. 2012. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence* 34, 7 (2012), 1409–1422.

37. Shaun K. Kane, Daniel Avrahami, Jacob O. Wobbrock, Beverly Harrison, Adam D. Rea, Matthai Philipose, and Anthony LaMarca. 2009. Bonfire: A Nomadic System for Hybrid Laptop-tabletop Interaction. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, New York, NY, USA, 129–138. DOI: http://dx.doi.org/10.1145/1622176.1622202

38. Rick Kjeldsen. 2006. Improvements in Vision-based Pointer Control. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility (Assets '06)*. ACM, New York, NY, USA, 189–196. DOI: http://dx.doi.org/10.1145/1168987.1169020

39. Rick Kjeldsen and Jacob Hartman. 2001. Design Issues for Vision-based Computer Interaction Systems. In *Proceedings of the 2001 Workshop on Perceptive User Interfaces (PUI '01)*. ACM, New York, NY, USA, 1–8. DOI:http://dx.doi.org/10.1145/971478.971511

40. Tiiu Koskela and Kaisa Väänänen-Vainio-Mattila. 2004. Evolution towards smart home environments: empirical evaluation of three user interfaces. *Personal and Ubiquitous Computing* 8, 3-4 (2004), 234–240. DOI: http://dx.doi.org/10.1007/s00779-004-0283-x

41. Georgios Kouroupetroglou, Alexandros Pino, Athanasios Balmpakakis, Dimitrios Chalastanis, Vasileios Golematis, Nikolaos Ioannou, and Ioannis Koutsoumpas. 2011. Using Wiimote for 2D and 3D pointing tasks: gesture performance evaluation. In *International Gesture Workshop*. Springer, 13–23.

42. Chiuhsiang Joe Lin, Sui-Hua Ho, and Yan-Jyun Chen. 2015. An investigation of pointing postures in a 3D stereoscopic environment. *Applied ergonomics* 48 (2015), 154–163.

43. Bruce D. Lucas and Takeo Kanade. 1981. An iterative image registration technique with an application to stereo vision. In *In IJCAI81*. 674–679.

44. I. Scott MacKenzie, Tatu Kauppinen, and Miika Silfverberg. 2001. Accuracy Measures for Evaluating Computer Pointing Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. ACM, New York, NY, USA, 9–16. DOI:http://dx.doi.org/10.1145/365024.365028

45. Rainer Malkewitz. 1998. Head Pointing and Speech Control As a Hands-free Interface to Desktop Computing. In *Proceedings of the Third International ACM Conference on Assistive Technologies (Assets '98)*. ACM, New York, NY, USA, 182–188. DOI: http://dx.doi.org/10.1145/274497.274531

46. Kenton O'Hara, Gerardo Gonzalez, Abigail Sellen, Graeme Penney, Andreas Varnavas, Helena Mentis, Antonio Criminisi, Robert Corish, Mark Rouncefield, Neville Dastur, and Tom Carrell. 2014. Touchless Interaction in Surgery. *Commun. ACM* 57, 1 (Jan. 2014), 70–77. DOI:http://dx.doi.org/10.1145/2541883.2541899

47. Ken Pfeuffer, Mélodie Vidal, Jayson Turner, Andreas Bulling, and Hans Gellersen. 2013. Pursuit Calibration: Making Gaze Calibration Less Tedious and More Flexible. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 261–270. DOI: http://dx.doi.org/10.1145/2501988.2501998

48. Alexandros Pino, Evangelos Tzemis, Nikolaos Ioannou, and Georgios Kouroupetroglou. 2013. Using kinect for 2D and 3D pointing tasks: performance evaluation. In *International Conference on Human-Computer Interaction*. Springer, 358–367.

49. Edward Rosten and Tom Drummond. 2006. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, Vol. 1. 430–443. DOI: http://dx.doi.org/10.1007/11744023_34

50. Lawrence Sambrooks and Brett Wilkinson. 2013. Comparison of Gestural, Touch, and Mouse Interaction with Fitts' Law. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration (OzCHI '13)*. ACM, New York, NY, USA, 119–122. DOI: http://dx.doi.org/10.1145/2541016.2541066

51. Matthias Schwaller and Denis Lalanne. 2013. Pointing in the air: measuring the effect of hand selection strategies on performance and effort. In *Human Factors in Computing and Informatics*. Springer, 732–747.

52. Garth Shoemaker, Anthony Tang, and Kellogg S. Booth. 2007. Shadow Reaching: A New Perspective on Interaction for Large Displays. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. ACM, New York, NY, USA, 53–56. DOI:http://dx.doi.org/10.1145/1294211.1294221

53. R. William Soukoreff and I. Scott MacKenzie. 2004. Towards a Standard for Pointing Device Evaluation, Perspectives on 27 Years of Fitts' Law Research in HCI. *Int. J. Hum.-Comput. Stud.* 61, 6 (Dec. 2004), 751–789. DOI:http://dx.doi.org/10.1016/j.ijhcs.2004.09.001

54. Rainer Stiefelhagen and Jie Zhu. 2002. Head orientation and gaze direction in meetings. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*. ACM, 858–859.

55. Radu-Daniel Vatavu. 2012. Point & Click Mediated Interactions for Large Home Entertainment Displays. *Multimedia Tools Appl.* 59, 1 (July 2012), 113–128. DOI: http://dx.doi.org/10.1007/s11042-010-0698-5

56. Eduardo Velloso, Jason Alexander, Andreas Bulling, and Hans Gellersen. 2015. *Interactions Under the Desk: A Characterisation of Foot Movements for Input in a Seated Position*. Springer International Publishing, Cham, 384–401. DOI: http://dx.doi.org/10.1007/978-3-319-22701-6_29

57. Eduardo Velloso, Marcus Carter, Joshua Newn, Augusto Esteves, Christopher Clarke, and Hans Gellersen. 2017. Motion Correlation: Selecting Objects by Matching Their Movement. *ACM Trans. Comput.-Hum. Interact.* 24, 3, Article 22 (April 2017), 35 pages. DOI: http://dx.doi.org/10.1145/3064937

58. Eduardo Velloso, Dominik Schmidt, Jason Alexander, Hans Gellersen, and Andreas Bulling. 2015. The Feet in Human–Computer Interaction: A Survey of Foot-Based Interaction. *ACM Comput. Surv.* 48, 2, Article 21 (Sept. 2015), 35 pages. DOI:http://dx.doi.org/10.1145/2816455

59. Mélodie Vidal, Andreas Bulling, and Hans Gellersen. 2013. Pursuits: Spontaneous Interaction with Displays Based on Smooth Pursuit Eye Movement and Moving Targets. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 439–448. DOI:http://dx.doi.org/10.1145/2493432.2493477

60. Nicolas Villar and Hans Gellersen. 2007. A Malleable Control Structure for Softwired User Interfaces. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction (TEI '07)*. ACM, New York, NY, USA, 49–56. DOI: http://dx.doi.org/10.1145/1226969.1226980

61. Daniel Vogel and Ravin 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 (UIST '05)*. ACM, New York, NY, USA, 33–42. DOI:http://dx.doi.org/10.1145/1095034.1095041

62. Edwin Walsh, Walter Daems, and Jan Steckel. 2015. An optical head-pose tracking sensor for pointing devices using IR-LED based markers and a low-cost camera. In *SENSORS, 2015 IEEE*. IEEE, 1–4.

63. John Williamson and Roderick Murray-Smith. 2004. Pointing Without a Pointer. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems (CHI EA '04)*. ACM, New York, NY, USA, 1407–1410. DOI: http://dx.doi.org/10.1145/985921.986076

64. Jacob O. Wobbrock, Kristen Shinohara, and Alex Jansen. 2011. The Effects of Task Dimensionality, Endpoint Deviation, Throughput Calculation, and Experiment Design on Pointing Measures and Models. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 1639–1648. DOI: http://dx.doi.org/10.1145/1978942.1979181

65. Xing-Dong Yang, Khalad Hasan, Neil Bruce, and Pourang Irani. 2013. Surround-see: Enabling Peripheral Vision on Smartphones During Active Use. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 291–300. DOI: http://dx.doi.org/10.1145/2501988.2502049