Behaviour–Aware Sensor Fusion: Continuously Inferring the Alignment of Coordinate Systems from User Behaviour

Anthony Steed[*], Simon Julier[†]
Department of Computer Science, University College London

ABSTRACT

Within mobile mixed reality experiences, we would like to engage the user's head and hands for interaction. However, this requires the use of multiple tracking systems. These must be aligned, both as part of initial system setup and to counteract inter-tracking system drift that can accumulate over time. Traditional approaches to alignment use obtrusive procedures that introduce *explicit* constraints between the different tracking systems. These can be highly disruptive for the user's experience.

In this paper, we propose another type of information which can be exploited to effect alignment: the behaviour of the user. The crucial insight is that user behaviours — such as selection through pointing — introduce *implicit* constraints between tracking systems. These constraints can be used as the user continually interacts with the system to infer alignment without the need for disruptive procedures. We call this concept *behaviour–aware sensor fusion*. We introduce two different interaction techniques — the *redirected pointing technique* and the *yaw fix technique* — to illustrate this concept. Pilot experiments show that behaviour–aware sensor fusion can increase ease of use and speed of interaction in exemplar mixed-reality interaction tasks.

Keywords: Mobile virtual reality, head-mounted display, 3D user interaction, selection tasks, augmented reality

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented and virtual realities H.5.1 [Information Interfaces and Presentation]: User Interfaces—Interaction styles I.4 [Image processing and Computer Vision]: Scene Analysis—Sensor fusion

1 INTRODUCTION

Recent developments in mobile technology have enabled the widespread deployment of mobile mixed reality (MR) systems. Mobile phones and tablet-based computers, equipped with sensing systems, graphics systems, networking, interaction and displays, boast capabilities that used to be limited to large, expensive and bulky desktop computers. Many MR applications now exist including games [1], sign translation [24] and identifying points of interest [12]. Many of these applications, however, only provide basic forms of display and very limited types of interaction. To achieve the next generation of mobile MR experiences, richer interactions must be supported.

In [29], we demonstrated how a complete VR / AR system, which consists of a stereo head mounted display, a head tracker and hand tracker, could be driven from a commercially available smartphone. The interaction was enabled through the use of IMU-based tracking. IMUs were used because they are — and will be for some time — the dominant form of tracking for mobile platforms. They are sourceless (can be used in a wide range of unprepared environments), low cost (only a few dollars per unit) and are very widely available (there are more than a billion smartphones in the world, many of which are equipped with IMUs [5]). However IMUs are known to drift over time and data from other kinds of sensors must be fused to mitigate this drift [3, 11, 27]. Nonetheless, as our empirical study in Subsection 3.2 shows, drift can continue to occur even in the presence of correction algorithms. As a result, periodic realignment is needed to reset this drift.

Existing approaches to alignment require the user undertake some specific task or action — such as orienting their head towards a known object in the environment [4] — to introduce known *explicit constraints* between the trackers. Given information about these constraints, the relative transformations between coordinate systems can be computed. However, there are two problems with this explicit approach. The first is that these interrupt the user and might disrupt engagement with a task or sense of presence in the environment. The second problem is that it would be difficult to judge when to perform the realignment.

In this paper we propose to use the *implicit constraints* which occur between coordinate systems as a result of *the natural behaviour of the user as they interact with the system*. Previous work has assumed that the user moves their head and hand independently of one another. However, many types of interaction require coordinated behaviour. For example, if a user points at an object with their hand we might infer that the hand is probably pointing at some point that the user is also able to see. The secondary insight is that tasks in the virtual environment, including operation of 3D user interfaces, are a good source of potential constraints from which we might infer relationships. For example, we might know that a certain set of actions happen in a sequence such as selecting a tool, then selecting an object, and thus the motions the user are likely to follow this pattern; from this we might be able to infer the alignment. We call this approach *behaviour-aware sensor fusion*. We use "fusion" to denote that we are using implicit information to make it possible to fuse data from one tracking system into another tracking system to estimate the relative transformation between each. To test the effectiveness of this approach, we implemented two techniques for behaviour-aware fusion and tested them on AR and VR scenarios using a mobile phone-based system. Our experiments show that behaviour aware fusion assists users in selection tasks.

The structure of this paper is as follows. We review the literature on tracking and alignment and mobile systems in Section 2. The problem statement is introduced in Section 3, and the challenges of using IMUs to perform interaction is illustrated by considering the drift between three identically-moved mobile phones. To address these problems, Section 4 describes our behaviour-based approach to sensor fusion and introduces two interaction techniques — redirected pointing (RPT) and yaw fix (YFT). Section 5 describes the experimental platforms used to evaluate these techniques. Sections 6 and 7 evaluate each technique in turn. It is show that participants report that RPT is easier to use, but there is no significant improvement in task completion times. YFT is both liked by participants and significantly reduces task completion times. We conclude by discussing limitations and possible extensions of the technique in Section 8.

[*]e-mail: A.Steed@ucl.ac.uk
[†]email: S.Julier@ucl.ac.uk

2 Background

2.1 Mobile MR Systems from Mobile Phones

The use of mobile systems for AR has a long history. The Touring Machine [10] is a seminal system in the field. The Touring Machine allowed a user to tour the campus of Columbia University and see geo-registered information such as the names of buildings. The system used a see-through HMD driven by a backpack-mounted computed. Many other systems, including Tinmith [21] or BARS [13], were developed using similar hardware configurations.

However, with the advent of smaller, high-powered computing devices, most recent AR research has focused on handheld devices such as mobile phones [28] or tablets. These devices provide a "through the device" AR mode where virtual graphics is overlaid on a video feed that is shown on the device. These platforms have become widely deployed, and numerous AR applications are now commercially available.

The recent development of technology such as Google Glass promises to reintroduce HMDs into the development of AR systems. To this end, in [29] we developed a complete MR / AR system based on the use of commercially-available mobile phone technology. Furthermore, this system provided full 3D orientation tracking of both head and hand. An issue then is to figure out how to support interaction.

2.2 3D User Interfaces and Mobile Devices

Interaction in 3D is a topic of considerable interest. 3D user interfaces are comprehensively covered in the book of Bowman et al. [8]. In this paper we are particularly concerned with the problem of how to support pointing at 3D targets in the environment to select them. Being able to select objects is fundamental task in MR systems, as selection is a common precursor to many other kinds of operations such as grabbing, moving, copying and deleting. While many techniques have been developed [6, 7, 20, 22], ray-based selection remains a popular choice because it is simple to implement, easy to use, and supports interaction at a distance. Other approaches, such as virtual hand techniques [20], require the user to be within arm's reach of the object they wish to select. Positioning the ray might be done in a number of different ways including user a cursor, touching on the display or ray-casting [9]. With other forms of AR such as hand-held AR, other techniques are possible. For example, Wither et al. compare selection on head-mounted and hand-held style AR systems [31]. We seek to do it in a way that provides natural interaction. With the types of trackers that our system uses, orientation is reported much more accurately than absolute position. Katzakis and Hori demonstrate that a mobile phone is a suitable controller for a 3D rotation task [15]. Thus we believe that ray selection, facilitated by orientation tracking, is an appropriate choice.

2.3 Tracking and Registration in Mobile Systems

Ray selection requires the tracking of both the head and the hand in 3D. However, errors and biases in the tracking systems can be frustrating and confusing, and can stifle natural user interaction. Given its importance, a great deal of research has been carried out into the development of tracking systems [30]. Arguably, one of the most significant advances in recent years has been the development of methods for the real-time visual tracking of natural landmarks on mobile computers [2, 16, 19, 23]. However, all vision-based methods make several key assumptions which we believe are not always appropriate in our application domain. The first is that the environment is populated by a sufficiently dense network of visual features. The second is that these features are often static — they are either fixed to unmoving infrastructure (such as buildings) or have fixed local topology (such as 2D markers). Finally, it is assumed that the MR system has sufficient computational resources available. However, the first two assumptions preclude the use of these techniques at night, when there is an obscurant (such as smoke) or in highly dynamic environments (e.g., crowds of people). They will also not work all the time as the user moves the camera around: a handheld device might be obscured from view. For the third assumption, when the mobile device is also being used to render dense 3D scenes, there might be insufficient resources available to support tracking. As such, we believe that low cost, lightweight, low processing power and standalone sensors are likely to still play a dominant role, either as a primary or secondary sensing system. In particular, given their cheapness and low computational cost, inertial measurement units (IMUs) are built into current smartphone, tablet and similar platforms. They are, and will probably remain for sometime, the most widely-used approach for tracking orientation in mobile MR systems.

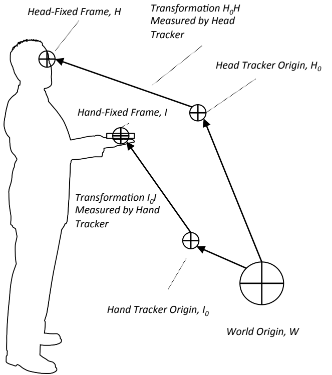
Head-Fixed Frame, H
Transformation H_0H Measured by Head Tracker
Head Tracker Origin, H_0
Hand-Fixed Frame, I
Transformation I_0I Measured by Hand Tracker
Hand Tracker Origin, I_0
World Origin, W

Figure 1: The kinematic chain of coordinate frames used to define the tracking system. Each frame is denoted by $\oplus$.

However, the use of IMU-only tracking introduces a number of challenges. IMUs normally integrate angular velocity which is measured by strapdown gyroscopes. Finite sampling rates together with noises and biases in the sensors mean these integrated estimates will drift over time [11]. Additional sensors, such as accelerometers and magnetometers, can be used to measure absolute references such as the direction of the local gravity vector and magnetic field. However, these are measured by sensors that have their own internal biases and noise sources. Furthermore, environmental effects (such as the presence of ferromagnetic material [3]) and movement of the user (acceleration of a user can be confused with gravity) limit the applicability of these sensors. Another issue is that many tracking systems are "black box" units, whose measurable behaviour includes the effect of an unspecified tracking or information fusion algorithm whose behaviours might — or might not — be reasonable. These difficulties are exacerbated when multiple devices are used at the same time, in which case they are operating under different local conditions and different patterns of behaviour.

One method to ensure alignment between the head and hand tracker is to use an *explicit* recalibration step in which the user manually aligns coordinate systems or strikes a specific pose with the user interface. For example, we could request that the user stand in a T-pose or look at and point at an object at the same time [4].

In summary, we believe that the effectiveness of mobile AR systems will be improved through the use of headmounted displays and support for more natural interaction. Current conditions mean that IMUs will be used, and ways to compute the transformations are required. We now look at the interaction problem and the effects of errors in detail.

3 Problem Statement

3.1 Tracking and Coordinate Systems

Fig. 1 shows the coordinate frames we consider. The world-fixed anchor frame is W. The head-fixed coordinate frame is H and the hand-fixed coordinate frame is I. To register the graphics with the real world, the transformation from the world to the head must be known. The head tracker measures the transformation from the head tracker's measurement origin (H_0) to the head. Using $\mathbf{C}_A^B(t)$ to denote the matrix transforms from frame A to frame B at time t, the transformation from the world to the user's head is given by

$$\mathbf{C}_W^H(t) = \mathbf{C}_{H_0}^H(t)\,\mathbf{C}_W^{H_0}(t)\,, \tag{1}$$

where $\mathbf{C}_W^{H_0}(t)$ is the base frame of the head tracker and $\mathbf{C}_{H_0}^H(t)$ is the measurement from the head tracker itself. The effects of biases, such as drift, can be modelled as a rotation on H_0.

Similarly, the transformation from the world to the user's hand is given by

$$\mathbf{C}_W^I(t) = \mathbf{C}_{I_0}^I(t)\,\mathbf{C}_W^{I_0}(t)\,, \tag{2}$$

where $\mathbf{C}_W^{I_0}(t)$ is the transformation from the world to the hand tracker base (which includes the effects of drift), and $\mathbf{C}_{I_0}^I(t)$ is the measurement from the hand tracker itself.

To support ray-based selection, we need to show the ray projected from the hand tracker in the user's head mounted display, which requires knowledge of the relative transformation between the user's head and hand, $\mathbf{C}_I^H(t)$. Using (1) and (2), its value is given by

$$\begin{aligned}\mathbf{C}_I^H(t) &= \mathbf{C}_{H_0}^H(t)\,\mathbf{C}_W^{H_0}(t)\,\mathbf{C}_{I_0}^W(t)\,\mathbf{C}_I^{I_0}(t) && (3)\\ &= \mathbf{C}_{H_0}^H(t)\,\mathbf{C}_{I_0}^{H_0}(t)\,\mathbf{C}_I^{I_0}(t)\,. && (4)\end{aligned}$$

Therefore, given $\mathbf{C}_{I_0}^{H_0}(t)$ and the measurements from the head and hand trackers, ray-based selection can be achieved. However, as explained above, the origin coordinate frames can be used to model the effects of sensor biases. As a result, $\mathbf{C}_{I_0}^{H_0}(t)$ can be both unknown and time varying. We illustrate this in an experiment which compares the performance of three orientation-only sensors.

3.2 Yaw Drift Example

We illustrate the presence of time-varying biases by comparing the computed heading of three iPhones over time. Note that we do not know the algorithm used by iOS to fuse sensor readings, but the APIs provide both separate acceleration, gyroscope and magnetometer readings, and an estimated device orientation. This device orientation can be retrieved via the CoreMotion API in a magnetic north coordinate system.

We attached three iPhone devices (two iPhone 4Ss, one iPhone 4) to a board and carried them on a walk around the local area. A plot of the yaw values reported by all three is shown in Fig. 2. All three start off with the same value. From 0–~135s, the devices are being carried through a building and down in a lift to the building exterior. In the period ~135–~180s, the two devices are placed against a calibration line (abutting a wall). From then until ~560s, the device is carried around the local area. This involves a complex route, but from ~365s to ~500s, the carrier is walking along a straight road. From ~560-~615s the device is placed on the same calibration line as previously. Then the devices are carried back up in a lift to their starting point.

We can see that the reported yaw value from the fused device orientation readings do show high-frequency changes that are very similar: this is likely due to the sensitivity of the gyroscopes. However on this short excursion the devices have diverged over time. In particular we can note that in the two periods when the devices are stationary they are diverging. In fact all three devices are moving in the first period: Device 1 drifts clockwise at ~0.06°/s, Device 2 drifts anti-clockwise at ~0.05°/s, and Device 3 drifts anti-clockwise at ~0.8°/s. At the end of this stationary period, the device have diverged by ~100°(Device 1 to Device 2), ~30°(Device 1 to Device 3) and ~70°(Device 2 to Device 3). Over the remaining time the absolute difference between the pairs varies up and down. A final note is that the divergences are greater in the second stationary period, but also no device reports the same orientation as the first stationary period. Device 1's discrepancy between first and second stationary periods is ~20°, Device 2's is ~150°and Device 3's ~25°.

This is a challenging scenario, but the situation in an urban environment would be a typical place of use for many mixed-reality systems. In [29], we reported a situation of yaw drift for an indoor scenario that matches our prototype one. In that case, with a device without a magnetometer, the divergences rates were higher.

While an analysis of the causes of drift and the biases of the sensors would be interesting, we leave that to future work. For the remainder of the paper, we assume that the tracking systems may drift over time, and thus a technique that can assist in aligning the coordinate systems at run-time has a broad applicability in parallel with other registration systems. This would be especially true for systems where a camera is not available or it would be difficult to justify the computational cost of other registration techniques.

4 Behaviour–Aware Sensor Fusion

4.1 Concept

Previous methods of calibration, which require users to look at or select specific targets, use obtrusive techniques to enforce a known value for $\mathbf{C}_I^H(t)$. Given the head and hand tracker measurements, (4) can be inverted to compute $\mathbf{C}_{I_0}^{H_0}(t)$. However, there are several difficulties with these explicit calibration techniques. The first is that these interrupt the user and might disrupt engagement with a task or sense of presence in the environment. The second is that it would be difficult to judge when to perform the recalibration. Perhaps the user could engage a calibration whenever they felt that the registration was incorrect, but ideally we would do it automatically when the system itself detected it was out of alignment. The third problem is that these recalibrations may be unnecessary when they are activated unless we observe the task the user is performing: for example recalibrating the hand tracker is probably not necessary if the user is performing a task where they are navigating an environment based on a travel in the direction of gaze metaphor and are not pointing at objects in the environment.

Given these limitations, we seek an *implicit* way to recompute the alignment. Rather than interrupt a user's normal interactions to force them to undertake a distinct recalibration process, we would like to develop methods which will allow the system to recalibrate the sensing systems as the user undertakes their normal interactions with the system. In particular, we believe that the patterns of behaviour that a user exhibits when they interact with a system introduces constraints on $\mathbf{C}_I^H(t)$. If these behaviours can be identified and classified, the tracking systems can be realigned without an explicit calibration step.

A simple example of a constraint would be to assume that users always tend to orient their hand to point in the direction in which they are looking. This would constrain the hand to be within a certain angle of the head. However, this is a poor constraint — the hand can point in a wide range of directions relative to the head. We believe that much better constraints can be derived by considering the way in which a user *interacts* with the environment. For example, if the user wishes to point at an object or grab it, it is highly likely that they will also look in that direction. Thus we can infer that if the user looks at an object, and also points in some direction, they may intend to grab that object, even if the pointing gesture is mis-tracked because of drift. We cannot know that they are looking at and pointing at the same object by simply casting rays or intersection volumes from the head and hand, because these might have diverged. However, we do have roll and pitch information that gives us some information. In particular there may be a correction to the

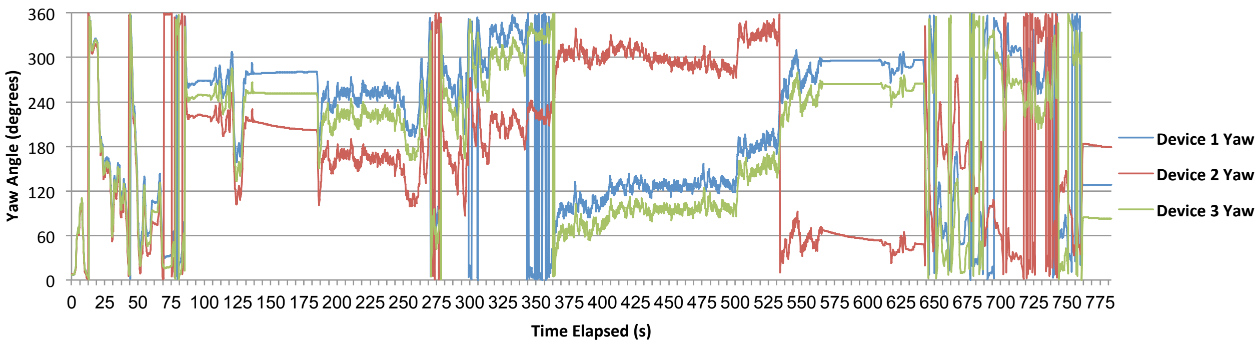

Figure 2: Time history of the yaw values computed by three iPhone devices held physically close to one another and carried through an urban environment. The difference between the devices shows the presence of both fixed misalignment offsets (between devices 1 and 3) and a time varying drift (device 2).

hand yaw that makes the ray from the hand hit the same object that was being looked at. If so, we might make that correction.

Similar observations can be made in other situations: if an object has a particular *affordance* [18], and the user is attending to that object, we might infer that is likely that any other actions on the object are to perform the action that is afforded, and thus we can interpret the tracking information coming from the hand. Affordances have previously been used to constrain animation and interaction (e.g. see [14]), but to our knowledge they have not be used as a source of potential constraints between coordinate systems.

Before introducing two techniques, we need to expand upon the definition of the coordinate frames described in Subsection 3.1. Our discussion so far has only considered the orientation of the head and hand tracker. However, the relative translation is required as well, but we do not have this information available. Rather, we assume the configuration shown in Fig. 3. The hand is offset in front of and below the head and can be in one of two positions — *Hand Up Position* and *Hand Out Position*. The former is used when the hand is raised to show a virtual UI controller. The latter is used when the user is interacting with objects in the world. Both offsets are made relative to the head position and take the head yaw into account. Thus they change position as the head rotates, staying in front of the user so that the virtual UI controller is visible.

4.2 Prototype One: Redirected Pointing Technique (RPT)

In this subsection, we describe the *Redirected Pointing Technique* (RPT). This derives its name from the redirected walking technique [25] in that it redirects tracking data to fit the interaction that would otherwise be difficult to perform. This technique provides a means by which drift in the yaw of the head can be compensated. Therefore, it only considers the case where the transformation between the head and hand need be maintained accurately and both together are allowed to drift with respect to the real world. This is typical in VR systems — because the user cannot see the real world, any drift in yaw will not be noticeable if it accumulates slowly. This is also true for some AR systems where registration is only achieved by considering the orientation of the user. This is widely used in many AR games for mobile devices [1]. Thus in this prototype we used a head tracker that comprises accelerometer and gyroscope only. The challenge is to register, implicitly, the hand into this head tracking coordinate system. Both trackers can be corrected by their accelerometers so that they diverge in only one degree of freedom (yaw).

The key assumption of the RPT is that the user will look at an object when they want to select it. Although this is not always guaranteed to be true, we believe it will be a valid assumption in most situations. The reason is that, because the user gets continuous feedback about the pointing direction from the ray that is drawn, and because they want to confirm the selection, they will usually look at the object. Further, if we didn't implement any correction, and in the presence of drift, over time users would probably learn that they need to look at where they are pointing more often so as to confirm that the coordinate systems had not drifted apart.

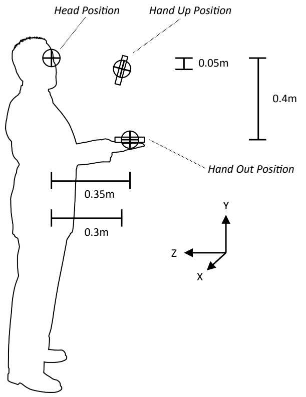

Figure 3: The relationships between tracker coordinate frames that are specified in order to create the user interface.

To achieve redirected pointing, we introduce the notion of a *pseudo hand orientation* which uses the pitch and roll from the hand tracker, but the yaw from the head tracker. The technique is visualised in Fig. 4. If the ray from the hand position with the pseudo hand orientation hits the same target as the ray from the head position with the head orientation, then we assume that the user is attempting to point at the target.

Mathematically, RPT is achieved by applying the constraint that $\mathbf{C}_I^H(t) = \mathbf{R}_H^I(t) = \mathbf{R}(roll_{hand}(t), pitch_{hand}(t), yaw_{head}(t))$ where $\mathbf{R}(\cdot,\cdot,\cdot)$ is the rotation matrix parameterised by the Euler angles. Because we assume there is no drift in the head tracker, $\mathbf{C}_W^{H_0}(k) = \mathbf{I}$. Therefore, substituting into (3), the head and hand tracker become aligned when

$$\mathbf{C}_W^{I_0}(t) = \mathbf{C}_I^{I_0}(t)\,\mathbf{R}_H^I(t)\,\mathbf{C}_W^H(t)\,. \qquad (5)$$

Because the drift is in a yaw direction only, $\mathbf{C}_W^{I_0}(t)$ only encodes a rotation about the yaw axis.

Some practical considerations must be made. The first is that head gaze is often not directly at objects, they may be off the centre of the screen, and the rays from the head and hand originate from different positions. Therefore, we use a type of volume selection

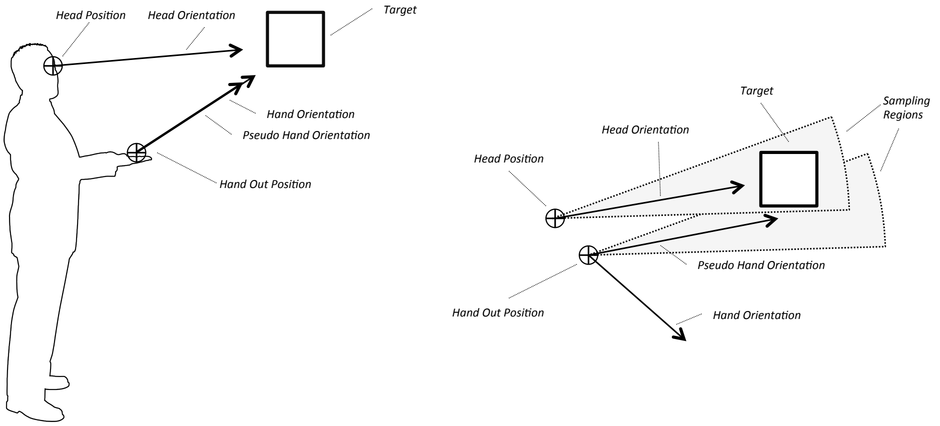

Figure 4: The definition of the pseudo hand orientation and its use in selection. Left: side view. The pseudo hand and hand orientation are aligned because the pseudo hand orientation uses the pitch and roll from the hand orientation. Right: top view. The pseudo hand orientation takes the yaw value from the head orientation. The sampling regions (cones) are used for ray hit detection.

for the pseudo hand orientation and head orientation hit detection. In particular, a conical bundle of rays are used to determine the probability of mutual selection of a target object by the head and pseudo hand. We use bundles that are 20° in diameter, but sample several times to find the most probable selection object. The cones are indicated as the *sampling regions* in Fig. 4.

A second consideration is that we don't want to constrain the user's hand too much. We want to give the user the freedom to pick a selection point on the object or use their hand to move away again. We have found that only applying the compensation to bring the hand within a few degrees of alignment of the potential target centre (10° being our current threshold) allows the user to select the object how they wish.

A final consideration is that we do not want to change the yaw compensation instantaneously. The yaw compensation is changed rapidly, at 60° per second with an ease in and ease out to avoid adding high frequency changes in orientation.

The technique bears some relation to snapping, but it is not a constraint that operates on one input stream. It is worth emphasising that it is not similar to gaze-directed selection, as the user must make a precise selection with their hand eventually. We have also found that it is difficult to activate accidentally, as it is actually uncommon that the head and pseudo hand point at the same target unintentionally. Even if the targets were distributedvery densely around the user, the technique would not automatically activate all the time because the pitch of the hand would still need to intersect the same target as the head. Section 6 examines this prototype in detail.

4.3 Prototype Two: Yaw Fix Technique (YFT)

Our second prototype fits the situation of use of MR system where it is important that the virtual environment is stable against the real environment. This is common in AR systems that reference geo-located objects. In this prototype the head-tracker uses an IMU that is registered against a magnetometer. In principle any registration technique could be used to stabilise this tracker; for example, as it is head-mounted it is more likely that a stable video could be retrieved. The challenge is to register a second, hand-held tracker into the same coordinate system. As demonstrated in Subsection 3.2, even if we use a device that has magnetometer calibration, the coordinate systems may still diverge.

To effect another form of behaviour–aware sensor fusion, our second prototype makes an assumption about the *temporal* ordering of actions. When a user wants to get access to information on the hand-held device, they will *first* raise their hand to see the device and *second* point towards the object they wish to select. This is a very reasonable assumption, but of course, we can't know that when raising their hand the user intends to look at it. However in the context of a task where the user needs to access information on a hand-held device in order to perform a subsequent pointing task that depends on that information, when they raise their hand it is quite likely that they will point afterwards and thus it is useful to realign the coordinate systems. Also in many environments, there are not many targets to point at directly upwards, so if the user raises their hand it is unlikely that they are selecting an object, in which case changing the hand yaw may have little impact. Of course, if there were targets above and the user pointed upwards, the implementation could not apply any adjustment.

As the user raises their hand, we can simply reset the hand yaw to use the head yaw. The correction is as described in (5). In this case we simply reset the angle, without a rate of change or an ease-in or ease-out. This is because the raise gesture can be quite fast, and also because the hand yaw is ignored when drawing the virtual user interface so that it always appears orthogonal to the viewer (see Subsection 5.2).

We refer to the technique as *yaw fix technique*. In [29], this is mentioned as an aside as an explicit technique for registering the coordinate systems, but not described in detail. In this paper we use it as an implicit technique. In particular, in the evaluation users were not told that it was operational. Section 7 examines this prototype in detail.

4.4 Clutch Technique

To act as a base-line for both evaluations, we implemented the *clutch technique*. This is an explicit approach used to realign the head and hand tracker. When pressing a select button, the hand tracker was disconnected from the virtual hand, alllowing the user to reorient their hand manually. This was implemented by measuring the change in orientation during each select button press, and then adding this as an offset to the yaw of the hand tracker origin. We found that the hand tracking was very stable in both systems and no user had problems with the dual use of the select button. In particular in the evaluations described later, and in informal trials, hand tracking was stable enough that during the action of pressing the select button, the hand rarely moved off the target due to jitter or jerk from the performance of the gesture. Thus selection was done with a quick press and no user was seen to have to hunt to find the target (hunting being the tactic of repeatedly pressing the button around the target).

The clutch was available in all experiment conditions. In one condition of both experiments only the clutch is available, so this condition will be referred to as the *Clutch-Only Technique* (COT) to avoid confusion.

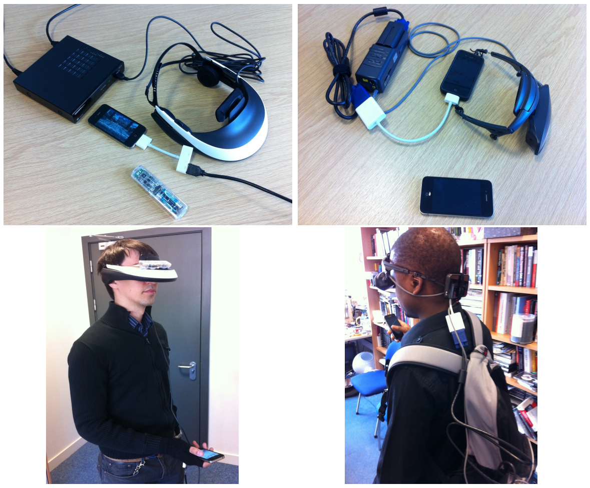

Figure 5: Top Left: VR system (Prototype One) components: Sony HMZ-T1 HMD and Hillcrest Labs Freespace tracker and an iPhone 4S generating video and acting as hand tracker. Top Right: AR system (Prototype Two) with an iPhone 4S attached to the Sony Glasstron and iPhone 4 as hand tracker. Bottom Left: User wearing the VR System. Bottom Right: user wearing the AR system. The HMD control box is placed in the outside mesh pocket of an empty backpack

5 Platform Overview

5.1 Hardware

Our platform was based on that of Steed & Julier [29] who recently presented a VR system based on an iPhone 4S. The first system described below is based on the same hardware. The second system used a variant of the hardware. The software was extended to support behaviour-aware sensor fusion, and also the new hardware configuration.

In both the VR and AR systems, an Apple iPhone 4S was used as the main device: it performed the main rendering and also received tracking data from one of two external devices. In the first system, the main iPhone was held in the hand and acts as renderer, input controller and hand tracker. We used a Hillcrest Labs Freespace Reference Kit, FSRK–BT–1 as a head tracker. The iPhone drove a Sony HMZ–T1 HMD. This was a stereo 1280 × 720 full pixel display in each eye with a horizontal field of view of 45°. We drove the HMD in mono. We used a standard iPhone HDMI adaptor. The equipment and a picture of a user using the system is shown in Fig. 5 Top Left and Bottom Left. The HMD is mains-powered and thus is not portable. It could be adapted to be portable.

In the second system, the main iPhone was mounted to the head on the back of the strap holding the HMD. The main iPhone acts as renderer and head tracker. A second iPhone was held in the hand. This second iPhone acted as a hand tracker and control input device. The main iPhone drove a Sony Glasstron LDI-D100BE, with 800 × 600 full pixel resolution in each eye and a horizontal field of view of approximately 28°. This HMD was stereo-capable but we drove the HMD in mono. We used a standard iPhone VGA adaptor. The HMD controller block was battery-powered and can be attached to a waist belt or placed in a backpack. The Sony Glasstron can be used in an optical see-through AR mode: there is a switchable panel at the back of the HMD that can be made opaque or partially transparent. The equipment and a picture of a user using the system is shown in Fig. 5 Top Right and Bottom Right.

The two HMDs could have been interchanged between the systems; the description above reflects the configuration used in the prototypes described in the remainder of the paper.

5.2 Software

The software architecture was as described in [29] but supporting later versions of the iOS SDK. It was written in a mixture of Objective-C and C++ for iOS 5.0.1 and iOS 6.1.2 using iOS SDK 5.1 (initial VR system) or iOS SDK 6.1 (later versions of VR system and AR system). The rendering software was written in OpenGL ES2.0 using modules from openFrameworks for model rendering and text rendering [17].

The two systems differed in their communication with the external devices. For the first system (the VR system), the Hillcrest Labs Freespace FSRK–BT–1 supported communication of sensor readings over BlueTooth but the necessary BlueTooth protocol was not directly supported by iOS. We used an open source user-space BlueTooth stack, btstack [26]. This required the main iPhone to be jailbroken. For the second system (the AR system), the two iPhones communicated using built-in BlueTooth, accessed through the iOS GameKit API.

The software system included a virtual hand-held controller. This virtual controller can show a number of virtual buttons inside the HMD view. The controller was operated by the iPhone that is held in the hand. In the prototypes in this paper, the virtual controller is very simple, and only supported a large selection button activated by tapping anywhere on the screen. When deployed for demonstrations the platform supports a broader range of functionality including navigation and editing, see [29].

The software used the tracking configuration as described in Section 3.1 and shown in Fig. 1. In order to define the origins of the tracking systems, we use the offsets as defined in Fig. 3. If the user raised their hand making a gesture as if bring their hand in front of their face, a large version of UI, see Fig. 8 Top Left, would be displayed at the Hand Up Position as defined in Fig. 3. Of course, we were not able to detect proximity to the face, so the gesture was recognised by the phone accelerating upwards and tipping up towards a vertical position (see [29]). When the user lowered their hand, a smaller version of user interface was displayed and the hand appeared to be at the Hand Out Position. Additionally a ray is drawn pointing out from the hand out position and this can be used to select objects, see Fig. 6 and Fig. 8 Top Right. The functionality of the user interface and its implementation is described in more detail in [29]. In the second prototype, additional information is displayed on the virtual hand-held controller when the hand is raised. See Section 7.1 and Fig. 8 Top Left. Note that for the AR system, we chose not to display the virtual controls on the hand-held iPhone, we showed them on a virtual display on the HMD. Although showing the controls on the hand-held iPhone might have made some sense as then there would be a one to one match with the hand movement, in practice we found it very difficult to control the lighting so that through the AR view, the screen, virtual graphics and real world were all visible. In future versions, we would prefer to use a smaller handheld device or even replace it by a glove or wrist-mounted sensor. Therefore, virtual graphics would be necessary anyway.

6 First Prototype and Evaluation

We performed a pilot study to judge the effectiveness of the Redirected Pointing Technique (RPT) for selection of objects in a VE. The RPT was compared against the Clutch-Only Technique (COT). The hypothesis was that participants using RPT would be faster than participants using COT at performing a sequence of selection tasks.

This experiment used the first system described in Subsection 5.1. Fourteen participants, all researchers and students in the UCL Computer Science department, undertook the experiment. All had experience with VR or AR systems, but had not used the equipment described in this paper before.

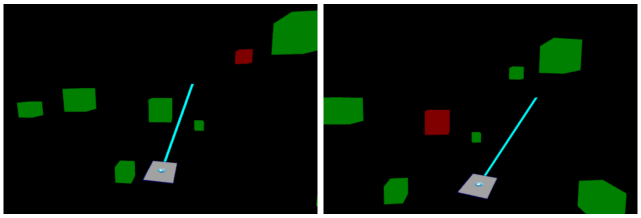

Figure 6: Selecting objects in the first evaluation. Left and Right, the user must select the red target object amongst a set of green objects.

6.1 Configuration

In this experiment, participants had to repeatedly find a target object in a set of objects and then select the target object. A set of 16 cube objects were placed in an arc around the user, at intervals of 12° covering 180°. This meant the user did not need to turn completely around, possibly tangling themselves up. Each target's size was chosen randomly in the range 0.3–0.6m width, and at a distance randomly chosen in the range 3.0-8.0m. This meant that the smallest target subtended approximately 2° of the field of view, but this was not found to be a problem for selection. The height of the object was randomly chosen between -1.5m below the head to 1.5m above the head. One target cube would be red, the rest green, see Fig. 6. The task was to select the red target object. The target object would then flash for one frame, turn green, and another target object would turn red. The targets were randomly chosen. Each trial consisted of 24 such selection tasks. For both trials for all participants the same random sequence was used. In the scene used in this experiment, consisting solely of a black background, cube objects and virtual controller, the rendering was performed at stable 60Hz. We injected a fake yaw drift at 2°/s in to the scene to exacerbate the problems with mis-alignment of head and hand. This did not start until the participant selected the first target.

6.2 Procedure

A within-subject design was used. Participants were naïve to the purpose of the experiment. Participants were informed that they would perform two selection trials, each consisting of a series of selection tasks. Half performed the trial using RPT first, followed by the COT, half using COT first followed by RPT. On putting on the HMD for the first time, the hand tracker was purposefully not aligned with the HMD so that the user could practise using the clutch mechanism. Note that the clutch is active in both RPT and COT trials. At this initial step they were facing away from the objects. Once they had practised using the clutch and were happy that it was now aligned with the hand held device, they were invited to turn to face the targets, find the red target object and proceed. None had problems following the instructions. The two trials took each participant no more than 10 minutes to complete. Participants were then de-briefed. First they were asked whether they could tell what the difference between the two trials was and how easy they thought the tasks were. Then we answered any questions that the participants had about the system. They were asked not to discuss it with colleagues for two days, by which time the experiment had been finished.

6.3 Results

The mean task completion time for RPT was 149.2s (std. dev. 54.3) and for COT was 163.9 (std. dev. 112.2). Although RPT had a marginally faster completion time, a one-way ANOVA indicated that this difference was not significant (p = 0.55). A two-way ANOVA considering COT and RPT as one factor and order of presentation as another showed the very significant impact of order of presentation with the second attempt being much faster than the first. For the first run users took a mean of 213.6s (std. dev. 91.0) versus a mean of 121.7s (std. dev. 53.1) for the second. This was highly significant (p = 0.0017).

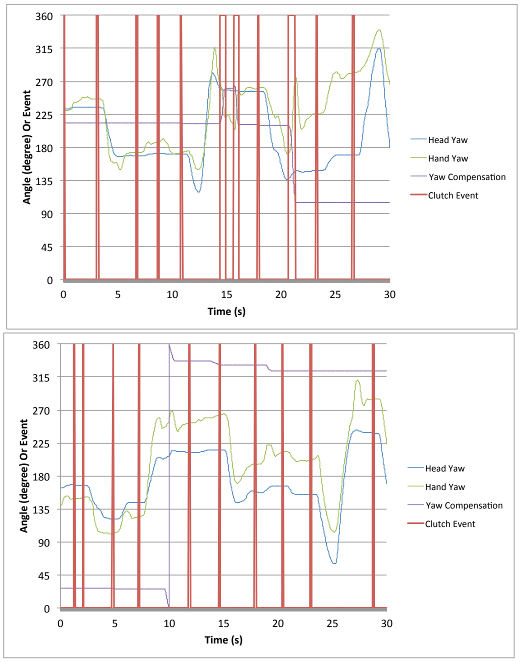

Figure 7: Excerpts of tracking logs for COT (Top) and RPT (Bottom). Each graph shows the head and hand tracking, the compensation between the two and the events when the user used the select/clutch button

Nevertheless when asked to comment on which of the two trials had been more difficult, participants either stated that they thought that was no difference (7 of 14) or that RPT was easier (6 of 14). One stated that he thought that RPT was trying to help and made it slightly harder to perform the task. It was noted that this participant was not moving their head very much at all and was only turning so that the targets were on the side of the screen, then pointing towards them. In this situation RPT could have hindered them because another target might have been in the centre of the screen. We didn't instruct participant to look at the objects and they were not aware how RPT worked. Two participants figured out that if they looked at objects in the RPT mode, selection was easier. One participant claimed that their hand was snapping to objects in the RPT mode. One suggested that the targets were somehow bigger in the RPT.

In both RPT and COT, but more so in COT, we observed participants making quite exaggerated gestures to point at objects. One participant barely used the clutch despite being aware of how to use it and demonstrating that they could use it. They were observed pointing the device towards their navel in order to select objects in front of them. Three participants mentioned that they thought they had to move their hand less in the RPT condition. Most participants indicated that, independent of the trial difficulty, that they performed better the second time because they knew what to do.

6.4 Discussion

Fig. 7 illustrates how the COT and RPT affect the head and hand rotation. The graphs show 30s examples from sessions with the head and hand yaw (with additional drift), the compensation that is applied to the hand yaw to effect interaction, and the times at which the clutch/select action was pressed (note that these examples were chosen at random from log files). The top graph shows COT and illustrates that the clutch is used infrequently (twice around 15s, and then around 21s). In the last of those uses, the clutch effects a

> 100°change in registration. For the RPT we see more frequent, but smaller changes in compensation that are applied automatically.

Whilst the results for task completion time were not significant, the feedback from users indicated that RPT may be a useful technique because it makes the task easier to perform. In retrospect, we felt that RPT might have been more useful if we had explicitly explained how it operated to the users. Other suggestions for development would be to do the redirect pointing in a different way: instead of correcting to perform the gesture, we can imagine observing which way a user had to turn their hand to select an object and assume that we should offset the tracker in the next few seconds to bring the hand in the opposite direction. This could be moderated by noting where on the screen the targets were.

7 Second Prototype and Evaluation

We performed a pilot study to judge the effectiveness of the Yaw Fix Technique (YFT) for selection of objects in a 3D AR system. Compared to the first evaluation, this experiments emphasises the potential of behaviour-aware sensor fusion in the context of a more complex task. The hypothesis was the participants using YFT would faster at performing a sequence of selection tasks than those using COT.

This experiment used the second system (see Subsection 5.1). Thirteen participants, all researchers and students in the UCL Computer Science Department, undertook the experiment. All had experience with VR or AR systems. Seven had used the equipment in the first evaluation. Although the average of their performance was faster than the six new subjects, this difference was not significant and the slowest completion times were recorded by a participant who took part in the first experiment. The experiment is a within-subjects design, and both groups of subjects showed the same trend, being faster on YFT than COT no matter the order of presentation.

7.1 Configuration

In this experiment, users had to repeatedly find a pair of objects, observe a visual instruction about which to select and then select the correct one of the pair. A set of 16 pairs of target cube objects were placed around the user, at intervals of 22.5° covering 360°. Unlike the previous trial, the user was expected to turn completely around as the equipment was mobile. Each target size was chosen randomly in the range 0.3-0.6m width, and at a distance randomly chosen in the range 3.0m to 8.0m. Each object in the pair was the same size and distance, but separated by 10° in the horizontal plane. One of the objects was cyan and the other yellow. The colours were randomly swapped depending on position. The height of the pair of objects was randomly chosen between -1.5m below the head to 0.5m above the head. At the start of each selection task the virtual UI controller would have placed upon it, a single cube with the colour of the object (cyan or yellow) that needed to be selected. The user thus needed to look at this controller before selecting one of the pair. They were instructed to find the pair of objects first and then raise their hand to see the virtual UI, see Fig. 8, Top Left. In the YFT condition this would realign the coordinate systems. However users were free to look at the virtual UI whenever they wished and some would look immediately after selecting one object to see the next colour to select. The participant then had to select the correct cube, see Fig. 8, Top Right. The object that was not intended to be selected would not react to selection events. When the correct object was selected, the pair of objects would disappear, two new objects would and the virtual UI would change to the colour of the target object in this pair. Each trial consisted of 24 such selection tasks. For both trials for all participants the same random sequence was used. In the scene used in this experiment, being an AR, the only objects drawn were the virtual UI, the selection ray and the pair of target objects. Fig. 8, Bottom Left and Bottom Right show views through the AR display during the task.

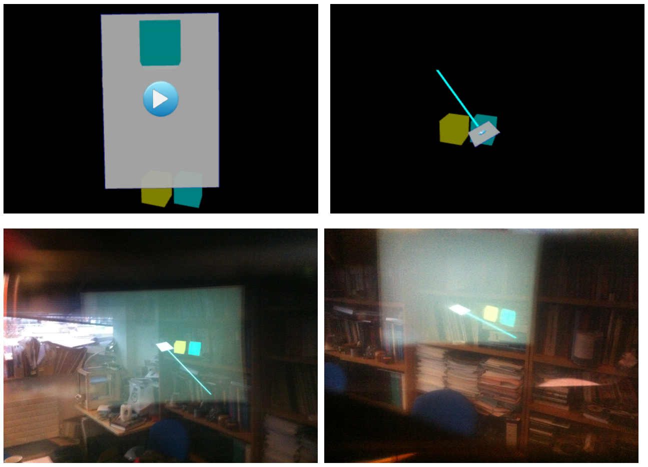

Figure 8: Selecting objects in the second evaluation. Top Left:. The user raises their hand to see the virtual UI indicating that they should select the cyan cube. Top Right: Lowering hand, the smaller UI and ray appear so the selection can be made. Bottom Left and Bottom Right: two views of similar situations photographed through the AR display. Note that the difficulty in making such shots detracts from the visual quality of the display.

As with the first evaluation, we injected fake yaw drift in to the scene to exacerbate the problems with mis-alignment of head and hand. This was done at 2°/s. The fake yaw drift did not start until the participant selected the first target.

7.2 Procedure

Many aspects of the procedure were similar to those in the first evaluation. Participants were naïve to the purpose of the experiment. They were asked to perform two selection trials, each consisting of a series of selection tasks, using two different behaviour-aware fusion mechanisms. In this trial, six performed the task using YFT first, followed by the COT, seven using COT first followed by YFT. As with the first trial, the system was not initially calibrated to allow the users to familiarise themselves with the clutch mechanism. Once they had practised using the clutch and were happy that it was now aligned with the hand held device, they were invited to turn to face the first pair of objects and commence the trial. They were asked to raise their hand to identify which of the yellow and cyan objects was the target and then select that target. This was repeated this until the trials were completed. There were no difficulties with following the instructions. The two trials took each participant no more than 12 minutes to complete. Participants were de-briefed and were asked whether they could tell what the difference between the two trials was and how easy they thought the tasks were. Then we answered any questions that the participants had about the system. They were asked not to discuss it with colleagues for two days, which allowed sufficient time for the experiment to be completed.

7.3 Results

The mean task completion time for COT was 325.0s (std. dev. 146.7s) and for YFT was 203.7s (std. dev. 46.7). A two-tailed paired comparison Student's T-Test indicated that the difference between the completion times was significant ($p=0.0048 < 0.05$). The difference in the means is quite large. Despite the differing order of the completion of the two trials, only 1 out of the 6 participants who completed COT second, managed to complete it faster than YFT (the expectation being that performance should increase over time). The impact of YFT is quite large, with the average improvement in completion time being a 29.7% reduction in completion time compared to COT.

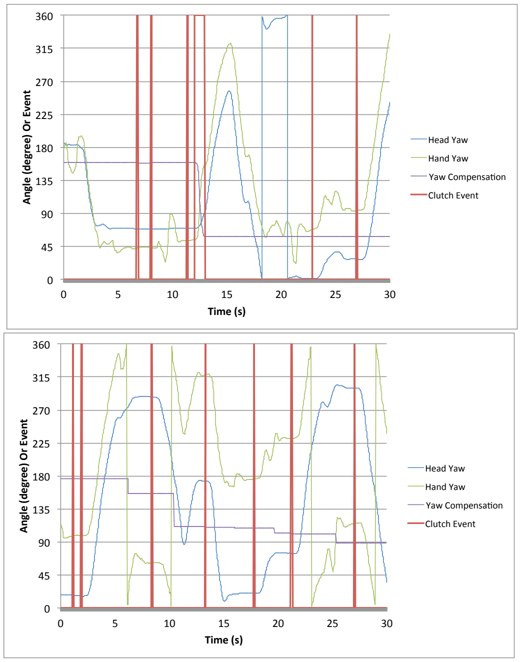

Figure 9: Excerpts of tracking logs for COT (Top) and YFT (Bottom). Each graph shows the head and hand tracking, the compensation between the two and the events when the user used the select/clutch button

When asked to comment on the difficulty, all but one (12/13) mentioned that the tasks were easier, or that they performed better in the YFT trial. Four made comments about the COT that they felt that the tracking became mis-aligned or the tracker was moved. One thought the two conditions were the same difficulty but they used the clutch more in COT; in actuality this participant was marginally (11%) faster in completing the tasks using YFT despite completing this condition first. In the COT condition one participant resorted to using both hands to manipulate the hand-held iPhone so that they could orient it correctly. One participant made the comment that they didn't use the clutch at all the first time (YFT), but lots the second time (COT). One stated that they found COT irritating (this was the second condition they completed). No-one realised what the difference between COT and YFT was.

7.4 Discussion

Fig. 9 illustrates how the COT and YFT techniques perform in a similar manner to Fig. 7 (Subsection 6.4). In the top graph we see COT and a single large change in compensation between the coordinate systems that the user effected manually using the clutch. For YFT we see frequent small changes, and as expected there is one change in compensation between most pairs of select events.

Compared to the first prototype and evaluation, the impact of the behaviour–aware sensor fusion is much clearer in this case, with YFT enabling significantly shorter task completion times. We should note that the visual quality of the Sony Glasstron, especially its transparency, is not as good as other AR HMDs. However, we should not expect the visual quality to differentially affect COT or YFT. No user had any problem seeing the target objects or ray.

8 Discussion and Conclusions

In this paper, we have considered the problem of developing unobtrusive techniques for aligning a tracking systems in order to support head and hand-based interaction for mobile MR applications. Rather than force users to undertake specific actions, we introduced the notion of behaviour-aware sensor fusion. This exploits the implicit constraints which occur between tracking systems as a result of the natural behaviours users exhibit as they interact with the system.

To test the approach, we created two different techniques which used two different constrains, and evaluated them in two experimental conditions. The first, the Redirected Pointing Technique, exploits the fact that, when a user attempts to select an object, they are likely to point their head and hand in the same direction. The second, the Yaw Fix Technique, exploits the likely temporal sequence of the gesture of raising hand to get information and then pointing at a target. Our results show both techniques lead to some improvements. Although the RPT did not improve trial completion time, some users reported that it made the tasks easier to perform. YFT significantly improved task completion times. Therefore, these initial results suggest that the behaviour aware fusion approach has the potential to be used effectively in other situations, but there are a number of issues which must be considered.

As with all methods which exploit conditions or constraints which are not measured, the performance of this approach largely depends upon the validity of the underlying assumptions. For this paper, we considered the problem of users selecting objects largely distributed on a plane. We chose to focus on selection because it is a common and fundamental operation, and the largely planar distribution of targets is accurate for many kinds of geo-registered MR systems. In our experiments, the underlying assumptions seemed to be valid and we did not encounter any cases in which the fusion constraints were applied inappropriately. However, in general this might not be the case, and there are several avenues of future research.

The first is to continue to analyse the performance of RPT and YFT in a wider range of application scenarios. For example, we are actively developing cultural heritage applications, in which users select and manipulate objects of the scale of houses. To this end, a large scale study of users could be carried out to identify where the assumptions might fail.

The second is to explore other types of implicit constraints. Fundamentally if the environment is designed to support a particular set of tasks, even if it is just exploration, we can decompose it into sub-tasks and then attempt to predict if the user is engaging in one of those sub-tasks. We might use patterns of interaction (e.g. movement of the head, eye scan-path, hand gestures, etc.) to create hypotheses about the state of the user within a sub-task. We also note that the techniques might be applicable in other areas of human-computer interaction, such as tangible interfaces, where motion data is increasingly used as a data source.

Others types of behaviour that one might look for are characteristic behaviours of users when the calibration is incorrect. We have noted that we observed users making very uncomfortable gestures in order to select objects. These included pointing towards their navel to select an object in front of them. In certain situations we saw the users making repeated motions to attempt to move the ray in one direction, but it travelled in an incorrect direction. For example, when the wrist is turned approximately 90°to the left or right, users appeared to sometimes pitch their hand when they should have yawed it, and vice-versa. This direction confusion is not something we have considered in this paper but we believe warrants further study as this and similar behaviours might indicate that coordinate systems are mis-aligned.

An important consideration for future work is whether behaviour-aware sensor fusion should remain implicit or made explicit. While we started off by stating that we wanted techniques that did not force the user to make explicit calibration steps, there is a mid-way position where we don't force them to make explicit calibration steps, but indicate which gestures or actions force calibration between the sensors as a side effect. We already saw from the two pilot trials that the users interpret the alignment in slightly different ways (e.g. thinking that the targets were larger). This type of technique might be more useful for expert users in that it can be learned as a side-effect of a specific action. Addressing this would likely require a longitudinal study and thus we would seek to integrate these techniques into a larger application framework that has more functionality.

Another important consideration is the likely relation to emerging techniques for registration with sensor combinations. We discussed the issues in Sections 2 and 3. For example, the magnetic field varies too much on a local scale to give a stable yaw constraint and GPS technologies are not going to progress to give accurate direction for a hand-held device. We discussed camera-based calibration, but noted, amongst other reservations, that the hand-held device is likely to be moving very fast, and will often not be looking at anything that can be tracked. If we assume that there are situations where camera-based calibration can be achieved, then behaviour-aware sensor fusion may still have an important role as a verification that the correct calibration has been achieved or is being maintained. Many constraint techniques are probabilistic in nature and/or based on optimisation techniques that are prone to local minima. Thus behaviour-aware sensor fusion might be useful technique to confirm a potential calibration.

The techniques we describe are already practical for our mobile VR/AR system that employs IMUs in its construction. This type of system set-up is likely to remain common for several years as the sensors are very cheap and are ubiquitous. While sensor technology will no doubt improve, we also note that sensor fusion technologies can utilise task knowledge to verify calibrations. In conclusion, we believe that behaviour-aware sensor fusion is an area that warrants further study.

References

[1] AppToyz. AppBlaster. http://www.apptoyz.com/shop/appblasterv2. Last visited August 1, 2013.

[2] C. Arth, D. Wagner, M. Klopschitz, A. Irschara, and D. Schmalstieg. Wide Area Localization on Mobile Phones. In *Proceedings of the 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 73–82, 2009.

[3] E. R. Bachmann, X. Yu, and C. W. Peterson. An Investigation of the Effects of Magnetic Variation on Inertial/Magnetic Orientation Sensors. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1115–1122, 2004.

[4] Y. Baillot, S. Julier, D. G. Brown, and M. A. Livingston. A Tracker Alignment Framework for Augmented Reality. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 142–150, Tokyo, Japan, 7–10 October 2003.

[5] S. Bicheno. Global Smartphone Installed Base Forecast by Operating System for 88 Countries: 2007 to 2017. Technical report, Strategy Anayltics, Boston, MA, USA, October 2012.

[6] D. A. Bowman and L. F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, I3D '97, pages 35–ff., New York, NY, USA, 1997. ACM.

[7] D. A. Bowman, D. B. Johnson, and L. F. Hodges. Testbed evaluation of virtual environment interaction techniques. In *Proceedings of the ACM symposium on Virtual reality software and technology*, VRST '99, pages 26–33, New York, NY, USA, 1999. ACM.

[8] D. A. Bowman, E. Kruijff, J. LaViola, and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional, Boston, 2005.

[9] A. Dünser, R. Grasset, and M. Billinghurst. *A survey of evaluation techniques used in augmented reality studies*. Human Interface Technology Laboratory New Zealand, 2008.

[10] S. Feiner, B. MacIntyre, T. Hollerer, and A. Webster. A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment. In *Proceedings of the 1st IEEE International Symposium on Wearable Computers*, ISWC '97, pages 74–81. IEEE Computer Society, 1997.

[11] E. Foxlin. Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter. In *Proceedings of IEEE Virtual Reality Annual Symposium VRAIS '96*, pages 185–194. IEEE Computer Society, 1996.

[12] W. GmbH. Wikitude. http://www.wikitude.com. Last visited 17th July, 2013.

[13] S. Julier, Y. Baillot, M. Lanzagorta, D. Brown, and L. Rosenblum. BARS: Battlefield Augmented Reality SystemBARS: Battlefield Augmented Reality System. In *In NATO Symposium on Information Processing Techniques for Military Systems*, pages 9–11, 2000.

[14] M. Kallmann and D. Thalmann. Modeling behaviors of interactive objects for real-time virtual environments. *Journal of Visual Languages & Computing*, 13(2):177 – 195, 2002.

[15] N. Katzakis and M. Hori. Mobile devices as multi-dof controllers. In *3D User Interfaces (3DUI), 2010 IEEE Symposium on*, pages 139 –140, march 2010.

[16] G. Klein. Parallel Tracking and Mapping on a camera phone. In *Proceedings of the 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 83–86, 2009.

[17] Z. Lieberman, T. Watson, and A. Castro. openFrameworks. http://www.openframeworks.cc/. Last visited 23rd April, 2013.

[18] J. Mcgrenere. Affordances: Clarifying and evolving a concept. In *Proceedings of Graphics Interface 2000*, pages 179–186, 2000.

[19] Metaio. Metaio SDK. http://www.metaio.com/sdk/. Last visited 30th July, 2013.

[20] M. R. Mine, F. P. Brooks, Jr., and C. H. Sequin. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 19–26, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[21] W. Piekarski and B. T. The Tinmith System — Demonstrating New Techniques for Mobile Augmented Reality ModellingThe Tinmith System - Demonstrating New Techniques for Mobile Augmented Reality Modelling. *Journal of Research and Practice in Information TechnologyJournal of Research and Practice in Information Technology*, 34(2):82–97, January–February 2002.

[22] I. Poupyrev, S. Weghorst, M. Billinghurst, and T. Ichikawa. Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. *Computer Graphics Forum*, 17(3):41–52, 1998.

[23] Qualcomm. Vuforia. https://www.vuforia.com/. Last visited July 27th, 2013.

[24] Quest Visual, Inc. Word lens. http://questvisual.com/us. Last visited 1st August, 2013.

[25] S. Razzaque, D. Swapp, M. Slater, M. C. Whitton, and A. Steed. Redirected Walking in Place. In *Proceeding of EGVE '02 Proceedings of the Workshop on Virtual Environments*, pages 123–130, 2002.

[26] M. Ringwald and P. Voser. btstack: A Portable User-Space Bluetooth Stack. http://code.google.com/p/btstack/. Last Visited 23rd April, 2013.

[27] D. Roetenberg, H. Luinge, and P. Veltink. Inertial and magnetic sensing of human movement near ferromagnetic materials. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 268–. IEEE Computer Society, 2003.

[28] D. Schmalstieg and D. Wagner. Experiences with handheld augmented reality. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 3 –18, nov. 2007.

[29] A. Steed and S. Julier. Design and implementation of an immersive virtual reality system based on a smartphone platform. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on*, pages 43–46, Mar. 2013.

[30] G. Welch and E. Foxlin. Motion tracking: no silver bullet, but a respectable arsenal. *Computer Graphics and Applications, IEEE*, 22(6):24 –38, nov.-dec. 2002.

[31] J. Wither, S. DiVerdi, and T. Hollerer. Evaluating display types for ar selection and annotation. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 95–98, 2007.