

Augmented Reality Working Planes: A Foundation for Action and Construction at a Distance

Wayne Piekarski and Bruce H. Thomas
Wearable Computer Laboratory
School of Computer and Information Science
University of South Australia
Mawson Lakes, SA, 5095, Australia
wayne@cs.unisa.edu.au, thomas@cs.unisa.edu.au

Abstract

This paper introduces the concept of augmented reality working planes to action and construction at a distance for mobile outdoor augmented reality systems. We have based our new AR working planes technique on CAD working planes, but by using coordinate systems relative to the body they can be specified and used much more intuitively than on a desktop system. We demonstrate in this paper how AR working planes can be used for the display of information, the manipulation of existing 3D objects, and the creation of new geometry. AR working planes are particularly well suited to supporting 3D modelling in mobile outdoor AR systems. This modelling task is a particularly difficult problem, because the environment is typically at a scale much larger than the user, and direct manipulation techniques can not be used.

1 Introduction

This paper defines a new theoretical framework, augmented reality working planes, to unify our action and construction at a distance interactions. We have previously implemented a number of individual action and construction at a distance techniques, such as infinite carving planes and orthogonal laser carving [23] [24] - but these were implemented in an ad-hoc way without a theoretical justification. This paper brings our previous work together under a unified framework, which is then used to indicate new possibilities which have not been explored yet. It also enables other investigators to extend these forms of interaction techniques in new directions, by providing a sensible means to precisely define new techniques.

Our framework is not designed for just outdoor AR, but also for general virtual environments such as indoor virtual reality (VR). While the existing 3D techniques described in the background section extend the user's interaction beyond arm's reach, they focus on operating at distances relatively close to the user where there are many cues available to accurately estimate relative position. We reference various studies to show that beyond a certain distance, the ability of humans to perceive depth is severely attenuated. This affects the accuracy of operations that can be performed at large distances, which is impor-

tant when interacting outdoors with structures that are beyond the depth perception capability of humans.

Our new AR working planes technique uses the projection of 2D cursors onto 3D planes to avoid the specification of depth values directly by the user. AR working planes is a combination and extension of CAD working planes [20] and image plane based techniques [12] [25]. The AR working planes technique only requires the use of 2D inputs and so can be implemented using a wide range of input devices, making it ideal for use in mobile environments. This paper describes the AR working planes concept in detail, discussing their placement in various coordinate systems and creation relative to the user or the world. The use of AR working planes to support action and construction at a distance is then described, including the manipulation of existing objects and the placement of vertices to create new geometry. Using AR working planes, the human is capable of performing interactions that are limited only by the accuracy of the tracking equipment in use and not by the human's lack of depth estimation capabilities.

The AR working planes technique is a fundamental component of the Tinmith-Metro modelling system [23] [24]. By combining AR working planes with a pinch glove based user interface, we have demonstrated the usefulness of AR working planes to perform real world modelling tasks. This paper describes the theoretical background which our modelling system currently partially implements. This modelling system interactively allows the user to create and edit 3D content in real-time while moving about the physical world. The user is able to use the position of their body and head, as well as a set of pinch gloves to interact directly with the environment. This is in contrast to the time consuming process of making all content changes indoors on a desktop machine and then uploading them to a mobile AR system. While scanning systems exist that are able to capture the geometry of existing objects, there is yet no equivalent tool that allows users to iteratively explore changes easily to designs while working outdoors. While there are some existing AR systems that provide very primitive modelling capabilities, they are by no means complete or fully understood. Our work in 3D modelling has provided us with the motivation

for the development of the AR working planes model to help us more thoroughly understand how to support action and construction at a distance techniques effectively.

2 Background

While 3D modelling in AR is relatively new, there has been quite a lot of research in the area of VR over many years. This research is useful because VR and AR share many similar properties but with varying degrees of reality and virtuality [17]. This section describes some previous modelling work that has been considered when developing AR working planes.

2.1 Action within arm's reach and objects within arm's reach

The simplest and earliest method of interaction in 3D environments is the use of direct manipulation, which is the closest approximation to the physical world possible. A very early example was by Clark [5], implementing direct manipulation of spline surfaces with a head mounted display (HMD) and a wand. Another example is Sachs et al.'s 3-Draw [29] which can create arbitrary models using direct manipulation of a stylus and tablet. Butterworth et al.'s 3DM [4] used direct manipulation to implement menus and dialog boxes to use with a 3D modelling system. Most immersive modelling systems (for example MultiGen's SmartScene [21]) implement this capability to provide both creation and editing of objects.

2.2 Action at a distance and objects at a distance

When objects are not within the reach of a user, action at a distance techniques may be used to allow the user to interact with the object. Liang et al.'s JDCAD [16] pioneered many new techniques such as lasers and spot lights using 3D input devices for selection and manipulation. Forsberg et al.'s work with apertures [12] extended Liang's spot lights to use a circular cursor on the hand projected from the head into the scene for selection. Pierce et al.'s image plane techniques [25] extended Forsberg et al.'s aperture projection concept to introduce a series of selection methods based on the projection of the user's hands and fingers onto the scene. It is important to realise though that these techniques are limited to only selection and manipulation, and cannot be used to create new 3D content since they are only 2D techniques.

Other non-linear cursor mapping techniques such as those by Mine [19], and the GoGo arm by Poupyrev et al. [26] were developed to provide other ways to move a cursor in 3D. These non-linear techniques rely on accurate depth perception to be effective, and this will be demonstrated later in this paper to be inadequate for modelling.

Bowman's CDS [3] implemented object extrusion based on intersecting points against a ground plane with a virtual laser beam to provide some early object creation capabilities.

2.3 Action at a distance and objects within arm's reach

Rather than interacting with the world at a distance, flying [28] and scaled world techniques [18] were developed to bring the entire environment within the direct reach of the user. Mine et al.'s CHIMP [18] implemented within arm's reach techniques based on proprioception and scaled world operations. Stoakley et al.'s Worlds-In-Miniature [30] demonstrated remote manipulation using small copies of the world held within the hands. Performing accurate modelling with these systems is difficult however because the user cannot match the size of the object to the size desired at normal 1:1 scaling.

2.4 Desktop CAD modelling

Computer Aided Design (CAD) systems have existed in various forms since the 1960s [32], and are used to create accurate 2D and 3D diagrams of real world objects. While keyboards allow the manual specification of any 3D point, devices such as mice are only 2D and cannot specify full 3D coordinates without extra assistance. CAD systems introduced the concept of working planes [20] that can be defined using numeric input, perpendicular to the view angle, or relative to the surface of another object. The 2D cursor is then projected onto the working plane to produce full 3D coordinates.

Working planes have features that can be explained best using an example of constructing a model of a house. Given a blank design, the 2D top down view is selected and a working plane is created at height zero. The user can specify a series of points to define lines forming the perimeter of the house. The user switches to a side view and then extrudes the perimeter up to create a solid shape. In-built objects supplied with the CAD system such as pyramids and wedges (or previous extrusion techniques) can be used to form the roof shape. Up to now this example has only used working planes that are perpendicular to the current view. The true power of working planes is most apparent when the view point cannot be aligned to the object. In this scenario, instead of using the coordinate system to specify working planes, an object facet itself can be used. To draw a window onto a wall of the house, the user nominates the wall and then simply draws against the surface. As each point is entered it is projected against the wall (working plane) and used to create a 3D vertex. If a picture is hanging on the wall, it can be moved along the surface of the working plane instead of on the plane perpendicular to the view point. Working planes act as a constraint mechanism that assists with the specification of three degrees of freedom using only a two degree of freedom input.

2.5 Mobile AR systems

Most AR research to date has been in the area of information presentation, with content created on desktop systems and then uploaded when required. Some example

systems are the Touring Machine [11], our first Tinmith system [22], BARS [14], the Context Compass [31], and some Studierstube systems [15]. In these systems, various interfaces such as tablets, mice, and tools are used to control the overall environment and the information display but not for general purpose 3D editing.

There are a small number of systems which are being used to perform true mobile AR object entry and editing operations. Work by Reitmayr et al. [27] used props to directly interact with 3D objects within arm's reach while indoors. Working outdoors is much more problematic than indoors however, since objects such as buildings and trees are not typically within arm's reach and at a scale much larger than the user. In our Tinmith-Metro papers [23] [24] we developed the infinite planes technique which allows the user to build up large solid shapes at a distance out of the intersection of many planes. The resulting shapes may then be carved iteratively to further refine the object. Baillot et al. [1] also demonstrated techniques that used the viewing of a vertex from two locations to triangulate its position in 3D. In these outdoor examples, AR object entry and editing was possible even at large distances from the user.

Although AR and VR systems share some similarities, AR is unique in that it requires the registration of the physical and virtual worlds. Therefore, flying around and changing the user's scale relative to the physical world is not possible. While existing action at a distance techniques that maintain the user's position and scale may be reused, other techniques are required to be able to arbitrarily create and edit 3D information within an AR environment. While the previously referenced AR modelling systems demonstrated some capabilities, only a small number of operations are supported. This paper describes a theoretical framework which can be used to explore all the useful techniques and the operations that are possible.

3 Limits in human depth perception

Humans gauge the distance to objects and their layout through visual cues acquired with the eyes, along with any other available senses such as sound, touch, and smell. The human sense of vision is unique in that it is capable of gathering information from a virtually infinite range of distances, whereas other senses tend to be useful only within close range. The retina in the human eye can be approximated as a 2D array of pixels (similar to a video camera) gathering light to produce a 2D image representing the 3D environment. While horizontal and vertical placement of objects in the image is easily obtainable, depth is ambiguous due to the flattened representation of the image, as depicted in Figure 1.

Depth information can only be estimated by analysing the contents of the images captured. The eyes and brain process a number of vision cues that occur in images to determine the depth positioning of objects in the scene, and are combined together to improve accuracy. Drascic

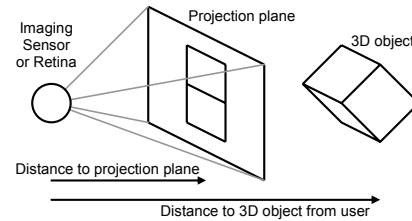


Figure 1 - 3D objects are projected onto a plane near the eye or an imaging sensor to form a 2D image

and Milgram [9] present a survey of perceptual issues in AR, discussing various depth cues and how mixed reality systems are limited in presenting them to the user. Cutting and Vishton [6] followed by Cutting [7] [8] provide detailed surveys on previous work in the area of perception and the determination of distance relationships between objects using visual cues. Cutting and Vishton collected results from a large number of previous studies and categorised nine cues, describing the range they are accurate over and the kind of depth information that can be extracted. These cues are occlusion, relative size, relative density, height in the visual field, aerial perspective, motion perspective, convergence, accommodation, and binocular disparity. Not all visual cues can produce absolute measurement information however; some cues can only provide relative ratios between objects or simple ordering information.

In normal daily life, the brain combines these cues together to produce situational awareness for the human. In VR and AR environments, some of these cues can be simulated with the use of head mounted displays (HMDs). HMDs can produce stereo images with offsets to match the distance between the eyes, and software can simulate fog and some environmental effects. While stereo HMDs give the user some feeling of depth perception, not all sensory information is generated and the brain may become confused by the inconsistencies presented.

To summarise the various cues and their effectiveness at different distances, Cutting and Vishton produced a graph depicted in Figure 2 that indicates the accuracy of each cue. This figure uses a log scale for distance along the X axis, and a normalised log scale along the Y axis with the smallest distance change measurable divided by distance. A value of 0.1 on the Y axis may indicate the ability to discern a 1 metre change at a distance of 10 metres, or a 10 metre change at a distance of 100 metres. Each of these curves is based on the data from numerous previously performed user studies and demonstrates that each cue is effective at different distances.

Cutting and Vishton defined three separate spaces around the body to categorise the depth estimation available. *Personal space* ranges from the body up to 2 metres, and is where humans perform most of their close up interactions, and so depth perception is highly refined here. From 2 to 30 metres is *action space*, where users may interact reasonably accurately with other objects (such as

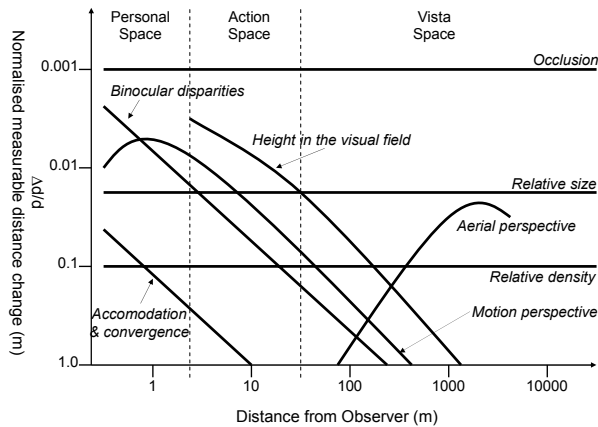


Figure 2 - Normalised effectiveness of various depth perception cues over distance (Adapted from Cutting and Vishton [6])

throwing a ball to hit a target), but with less accuracy than personal space. Beyond 30 metres is *vista space*, where objects appear flat and distance estimations become quite poor in comparison. Figure 2 shows where these three spaces are located relative to the accuracy curves previously described.

It appears that a human's ability to reconstruct 3D information about a scene is most capable when operating within close range to the body. Since humans mainly deal with objects that are within arm's reach, this sense (named proprioception) is highly refined and was used by Mine to improve user interfaces for 3D environments [18]. At larger distances however, these abilities attenuate very rapidly to the point where beyond 30 metres or so it is difficult to perceive absolute distances. When modelling large outdoor structures such as buildings, distances of 30 metres or greater are quite common. If distances cannot be perceived accurately for the modelling tasks required, then performing action and construction at a distance operations will require extra assistance to be useable.

4 AR working planes

Previously described techniques such as working planes [20], selection apertures [12], and image planes [25] were developed to project the locations of display-based cursors onto a 3D environment. These techniques are useful for selection and manipulation operations in vista space because there are no restrictions on the range of use, and the techniques are just as easy to use within arm's reach or kilometres away. Image planes and selection apertures are not capable of specifying distance however, since the plane from the view frustum is used for the cursors and depth is not required to be resolved. Assuming a typical perspective projection, the accuracy of vertical and horizontal motion in all of these techniques is proportional to the size and distance of the object, but attenuates at a constant rate less than that of human depth perception. With the use of HMDs, the cursor is represented using pixels

and introduces a pyramid of uncertainty specified by the pixel size at the projection plane. This also means that when an object becomes far enough away, it will eventually not be visible within a single pixel. An important property of interactive modelling is that users can only perform manipulations that are visually verifiable. There is no need to provide the user with the capability to move a mountain on the horizon 5 metres to the right because the change is not visually noticeable. Only by approaching the object will the user notice any accuracy problems, and these can then be corrected since it is a change that can be verified. Based on this argument, the use of projection techniques imposes no accuracy limitations noticeable by the user.

Using the previously discussed projection concepts, these can be extended into the AR domain to perform interactive modelling outdoors. We therefore propose a concept named *augmented reality working planes* that is based on the working planes concept used in traditional CAD systems. AR working planes can be created in the environment relative to the user or other objects, and stored in one of four possible coordinate systems. These planes can then be used as a surface to project a 2D cursor on to, resolving full 3D coordinates to manipulate existing objects and create new vertices. Since planes are by definition infinite in size, the user can project the cursor onto the plane from almost any location, although the accuracy the user will achieve decreases as the plane becomes parallel to the user's view direction. AR working planes improves on existing image plane-based techniques because the plane can be located anywhere in 3D space, allowing the calculation of depth at any distance and interaction in all three dimensions. This technique is also a mobile alternative to desktop CAD systems because the 3D view and working planes can be specified using the body in the physical world. To control the cursor projected against the AR working plane, any 2D input device can be used. The cursor is projected onto the surface of the plane and so no depth information is required, allowing a wide range of input devices to be used.

The use of AR working planes does impose some limitations on the user, and requires them to specify distance by creating a plane and then drawing against it from a different direction. Two operations from separate locations and orientations are usually required so that depth can be extracted without requiring the user to estimate it. While previous research has shown that decomposing 3D tasks into 1D or 2D units was not efficient [16], in the scenario of working in vista space there is no alternative. To support this argument, Ware [33] and Hinckley [13] both state that reducing degrees of freedom is useful when it is hard to maintain precision in certain degrees while adjusting others. In vista space, depth estimation is poor and so removing this degree of freedom is the best option to prevent the user from introducing further inaccuracies.

5 Coordinate systems

Since people think in terms of objects relative to their body rather than abstract coordinate systems and view points, our AR working planes are created using the user's body to make them as intuitive as possible. An important improvement is that these AR working planes can be created and fixed to a number of coordinate systems that humans easily understand.

Using these concepts, a number of different coordinate systems can be identified that are suitable for performing modelling tasks, as depicted in Figure 3. We use the common terms world, location, body, and head to describe these coordinate systems. In Figure 3, the user operates in a world coordinate system that is anchored to some fixed point in the physical world. Using a positioning device, location coordinates are measured relative to world coordinates and represent the location of the user's feet but with the direction fixed to true north. Using an orientation sensor mounted on the hips for example, body-relative coordinates can be calculated by applying an offset to transform from the feet to the hips and then applying the orientation. Head-relative coordinates are similarly calculated with the appropriate height and orientation of the user's head. The height values used for body and head coordinates can be either measured once and stored as a constant, or captured from a tracking device. We do not associate a coordinate system with the hands, because the hands are used to supply the cursor which moves across the AR working plane. Other coordinate systems are also possible, assuming appropriate tracking is available, but the identified coordinate systems are the main ones humans intuitively understand.

Information can be stored relative to any of the coordinate systems described in Figure 3. The surround-fixed windows by Feiner et al. [10] map to body-relative, display-fixed windows map to head-relative, and world-fixed windows map to world-relative. Our coordinate systems are similar to the head, body, and world stabilised information described by Billinghamurst et al [2]. The menus and tool palettes floating about the user implemented by Mine and Brooks [18] map to body-relative. Using the coordinate systems defined here, we extend existing previous work to include not only the presentation of information, but also the placement of AR working planes so that points may be created and objects manipulated at a distance. In this section, we describe AR working planes that have been created relative to each of the four described

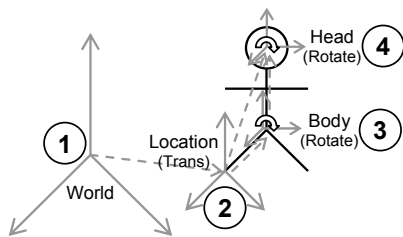


Figure 3 - Coordinate systems used for the placement of objects at or near a human

coordinate systems. Although body-relative coordinates are described here, they are not implemented in Tinnith-Metro because we do not currently require their use. All other coordinate systems are used in our implementation however, since they all have particular uses. In this section, figures are used to show the effect on each AR working plane of body translation, head rotation, body rotation, and combination movements in the environment.

5.1 World-relative coordinates

World coordinates are the top-level coordinate system used to represent positions over a planet or other large areas of interest. Objects that are specified relative to the origin of the world coordinate system are anchored to a fixed place in the physical world, and are completely independent of the user's motion, as depicted by (1) in Figure 3. In virtual environments, most objects are created world-relative since they are not attached to the user and may move independently, with examples being buildings, trees, and automobiles. The user's coordinate systems are also specified in world coordinates, since their position and orientation are returned from tracking devices that are world-relative. Figure 4 depicts a user moving in the environment with the AR working plane remaining fixed since it is in coordinates independent of the user's translation, head orientation, and body orientation. World-relative AR working planes are commonly used when working with buildings and the user desires to keep the planes fixed relative to the walls at all times.

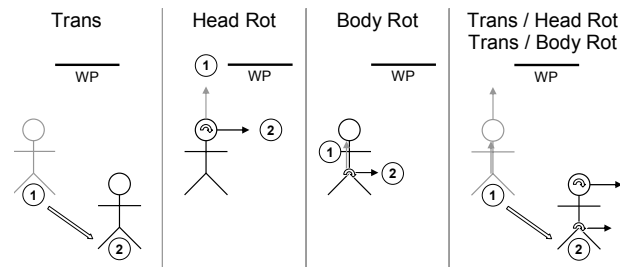


Figure 4 - World-relative AR working planes remain fixed during user movement such as translation and rotation

5.2 Location-relative coordinates

Location coordinates are derived by taking the current position of the user from a tracking device and adding this to the origin of the world coordinate system. The axes for both location and world coordinates are still aligned except there is a translation offset between the two, as depicted by (2) in Figure 3. With location coordinates the orientation of the user has not been applied, and so any changes in rotation will have no effect. An object placed in location-relative coordinates will always appear at the same true compass bearing from the user and maintain the same distance during motion. Location-relative coordinates are particularly useful for displaying an immersive compass to the user - the compass labels are attached around the user at a fixed radius and stay at the same ori-

entation no matter what direction the user is looking. Another use is to attach a virtual camera at a fixed distance and direction from the user at all times, which follows the user's location but does not move with head or body rotation. Figure 5 depicts the effects of user motion on an AR working plane that is location relative, where the plane moves with the user around the world. With user translation the plane moves with the same transformation, but any rotation has no effect. The main uses for location-relative coordinate systems are the placement of vertices and object manipulation at fixed orientations. These fixed orientations are useful when working with buildings, keeping the AR working plane parallel to the walls but still moving relative to the user.

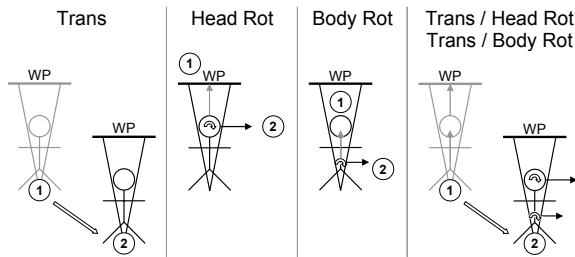


Figure 5 - Location-relative AR working planes remain at the same bearing from the user and maintain a constant distance

5.3 Body-relative coordinates

Although it is possible to define any number of coordinate systems, in many cases it does not make sense to create objects relative to arbitrary parts of the body. A user's sense of proprioception is focused about its main components such as the hips and the head, and so these will be the main focus. Body coordinates are defined relative to location coordinates except that orientation of the hips is added, as depicted by (3) in Figure 3. Objects placed in body-relative coordinates will always appear in the same location relative to the hips as the user moves around, with a good example being a tool belt worn by a worker. When walking around or when moving the head, the tool belt always remains in the same fixed position, ready to be accessed by the hands. Body-relative differs from location-relative in that the rotation of the hips affects the attached objects, whereas location relative ignores any rotations by the user. The cockpit of an aircraft is also similar, where controls are always at the same location relative to the user's hips but the aircraft can fly around and keep the controls at the same locations. Figure 6 depicts the effects of user motion of the body on an AR working plane that is body-relative, where the plane is attached to the hips of the user as they move around the world. Although body coordinates are very intuitive within arm's reach due to proprioception, they become more confusing at further distances since extra visual inspection is usually required. Some possible uses for body-relative coordinate systems are keeping interaction widgets nearby for easy access, and the display of non-critical status information.

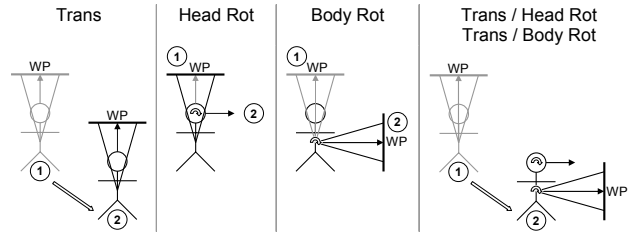


Figure 6 - Body-relative AR working planes remain at a fixed orientation and distance to the hips and are not affected by any rotation of the user's head

5.4 Head-relative coordinates

Head-relative coordinates are similar to body-relative in that they add rotations to the location-relative coordinates, and can be defined relative to either location or body coordinates, as depicted by (4) in Figure 3. The only difference between head-relative and body-relative coordinates is the part of the body that the information is attached to. Objects placed in head coordinates will always appear in the same location relative to the user's head, with a good example being a floating status indicator on a HMD, which is always visible at the same location. Figure 7 depicts the effects of user motion of the head on an AR working plane that is head-relative, where the plane is attached to the head of the user as they move through the world. When the user moves in the world, the plane will be translated and rotated to remain fixed within the field of view. The main use for head-relative coordinate systems is the placement of display status information and object manipulation. Head-relative mode is the most natural choice for object movement since it allows the user to adjust all three degrees of freedom by moving the body.

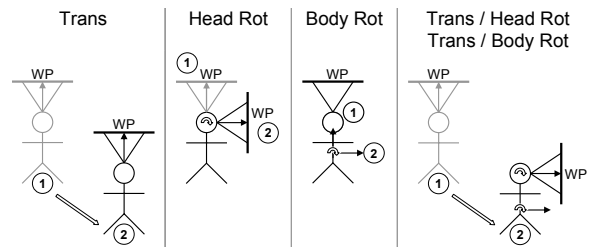


Figure 7 - Head-relative AR working planes remain attached to the head during all movement, maintaining the same orientation and distance relative to the head

6 Plane creation

In order to take advantage of AR working planes, the plane must first be created in the environment. During creation, AR working planes must be located in one of the coordinate systems defined earlier, which will affect the operations that can be performed. Planes are typically defined using the surface equation $Ax + By + Cz + D = 0$, where (A, B, C) is the normal vector and D is calculated using any point on the surface. This section describes different methods for creating planes that may then be used for manipulation and vertex creation. In each case, it can

be assumed that V_{up} points toward the sky, and V_n is the direction the user is viewing. We define the V_{up} vector so that planes which use it will always be vertical and invariant to the pitch or roll of the user's head. The five creation descriptions in this section define how the point on the plane and the plane normal P_n is specified.

6.1 Created along head direction

Figure 8 depicts a user creating a plane originating from their head, parallel to the direction that the head is viewing. If the user is viewing in the direction of true north, then the plane will be infinite in the north and south directions, with east and west divided by the plane. Constraints may be applied to the orientation of the head so that only some degrees of freedom are used to create the plane, such as being perpendicular or parallel to the ground. For the perpendicular case, the normal may be specified mathematically with $P_n = V_n \times V_{up}$, and with the user's current location used to calculate D . Since AR working planes are only useful when facing the user for cursors to be projected onto it, the user must be able to move independently of the plane to new viewing locations. This method is therefore only relevant with world-relative coordinates since the plane is decoupled from the user's motion.

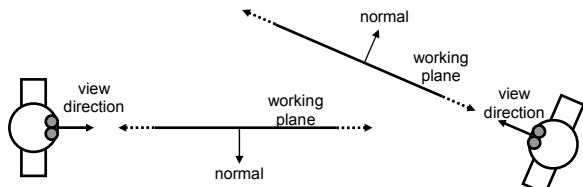


Figure 8 - AR working plane created along the head viewing direction of the user

6.2 Created at offset with user's head direction as surface normal

Figure 9 depicts a user creating a plane that is located at a fixed distance away and with surface normal matching the user's view direction. If the user is viewing in the direction of true north, the plane will have a surface normal pointing north and be infinite in the east and west directions. Constraints may be used to restrict the degrees of freedom of the orientation of the head for creating the plane. For a plane perpendicular to the ground, the plane normal may be represented as $P_n = V_{up} \times V_n \times V_{up}$, and to calculate D we use the current location and add to this an offset vector $P_n \times d$, where d is the distance in metres for the plane. Since the plane is facing the user it is ready to draw on and is suitable for use with all coordinate systems defined previously. The limitation of these planes is that the distance from the user must be specified with another input method, and the user may not be able to perform this accurately. This does allow the user to specify a virtual glass plane in front of them to attach AR information on however.

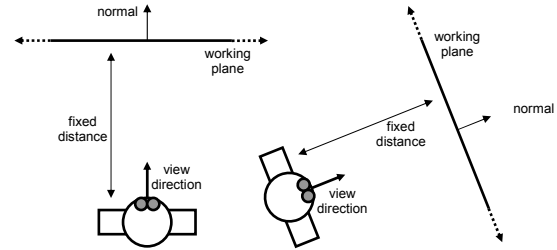


Figure 9 - AR working plane created at a fixed offset and with the surface normal matching the view direction of the user

6.3 Created at an object with user's head direction as surface normal

This technique is very similar to the previous in that the plane's surface normal is based on the user's view direction. The difference is that the plane is created so that it passes through the intersection point a user has selected on an object in the world. The plane normal is found using $P_n = V_{up} \times V_n \times V_{up}$ to match the user's heading (but not pitch and roll), and the intersection point on the object's surface is used to calculate D . Figure 10 depicts a user creating a plane at the intersection point of an object. These planes are most useful when created in head-relative coordinates for manipulation operations, although any other coordinate system is also possible.

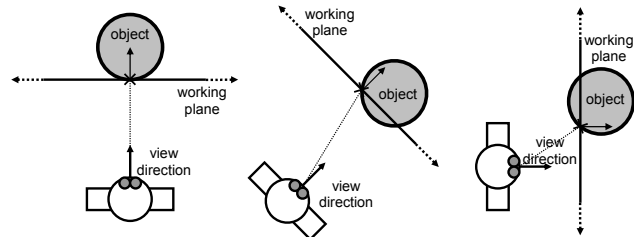


Figure 10 - AR working plane created at intersection of cursor with object, and normal matching the user's view direction

6.4 Created aligned to an object's surface normal

Figure 11 depicts a plane created to match the surface of a nominated facet on an object. Each of the objects has an AR working plane that is coincident with the selected facet, making it invariant to the user's current position and orientation. As long as the object facet is visible and can be selected, then it can be used to spawn an AR working plane in the environment. These planes are created by simply copying the surface properties of the selected polygon to find the plane equation. Since the plane is immediately visible to the user when created, it is ready to draw on straight away and is suitable for use with all coordinate systems. World coordinates are the most logical usage however, since the planes are defined relative to an object that is typically in world coordinates. Uses for other coordinate systems are discussed in the next sections.

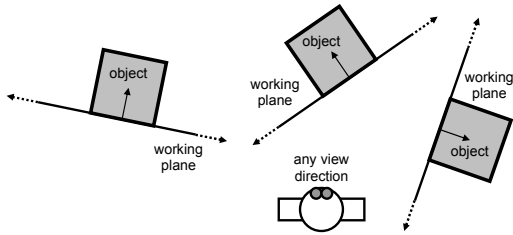


Figure 11 - AR working plane created relative to an object's surface

6.5 Created at an intersection point using another object's surface normal

Using a similar technique to that discussed previously, the facet of an object may supply a surface normal for an AR working plane created at another object. Figure 12 depicts a plane created at the point where the user's cursor projection intersects an object in the environment. The surface normal is copied from an object selected previously with the same method. Therefore the new plane will use the normal of the first selected facet for P_n , and the point used to find D will be the second intersection point selected. This technique is useful for manipulating objects relative to the surfaces of others and so is the most logical with world-relative coordinates, although other coordinate systems are possible as well.

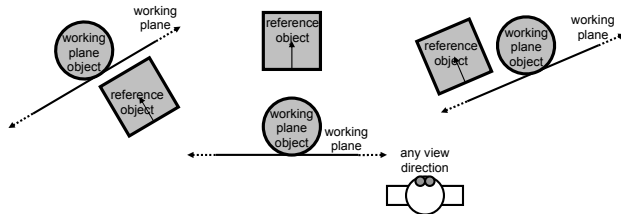


Figure 12 - AR working plane created at a nominated object based on the surface normal of another reference object

7 Object manipulation

Given the ability to place down AR working planes in the environment, one possible use is the implementation of translate, scale, and rotate operations using a 2D input device. The first step is to create a working plane in the environment using one of the previously described techniques relative to an appropriate coordinate system. An important consideration is ensuring that the working plane will be facing somewhat towards the user to improve accuracy - planes with a high aspect ratio will introduce large depth changes with only small 2D motions. The choice of coordinate system determines the type of operations that can be performed. When using AR working planes in head coordinates, the results share similar properties to selection using image planes [25].

Translation operations where the object is accurately moved across the AR working plane surface can be performed as shown in Figure 13. Two points are projected onto the plane and are used to calculate a translation. This translation is then applied to the object to move it to the

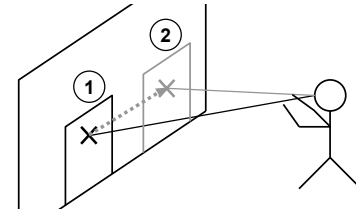


Figure 13 - Manipulation of an object along an AR working plane surface

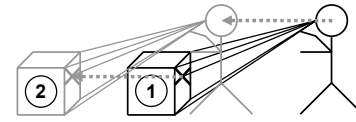


Figure 14 - Depth translation from the user moving a head-relative AR working plane

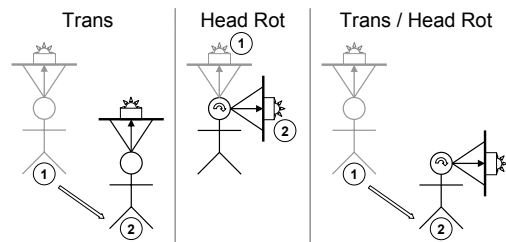


Figure 15 - AR working plane attached to the head can move objects with user motion

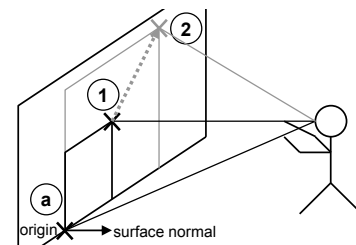


Figure 16 - Scaling of an object along an AR working plane with origin and two points

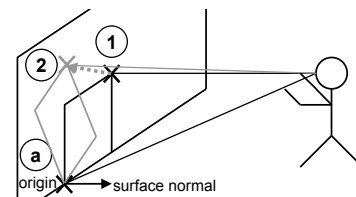


Figure 17 - Rotation of an object along AR working plane with origin and two points

desired location, with the offset always being along the surface of the plane. If the AR working plane is attached to location, body, or head coordinates then varying the user's position will drag the object around, as depicted in Figure 14. When using body or head coordinates, translations and rotations can be combined together, such as depicted in Figure 15. By combining the 3D motion of the body with 2D cursor motion, complex manipulation in 3D can be easily performed. Since the user is able to understand the local movements of their body, accurate translations can be made on distant objects.

Scaling and rotation operations can be performed along the surface of an AR working plane and requires three input points – an origin for the operation, and two points to specify either a direction and magnitude vector, or a rotation angle. Figure 16 depicts the cursor points to calculate a scaling transformation, and Figure 17 depicts the cursor points to calculate a rotation transformation.

8 Vertex placement

The second more novel use for AR working planes is the placement of new vertices in the environment. Selection and manipulation of existing objects has been implemented previously using a number of techniques, but there is still a lack of techniques for the creation of new geometry at a distance. Figure 18 depicts how a user can project a 2D cursor against an AR working plane and create vertices anywhere on the surface. Similar to the previous object manipulation section, this operation can be performed using an AR working plane in any coordinate system and created using any technique.

Apart from just creating points against fixed surfaces, if the AR working plane is relative to head coordinates then it will move with the motion of the user, as depicted in Figure 19. As the user translates and rotates, the AR work-

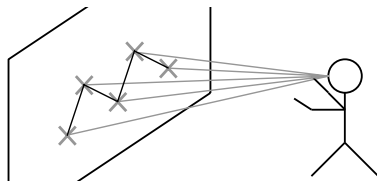


Figure 18 - Vertices are created by projecting the 2D cursor against an AR working plane

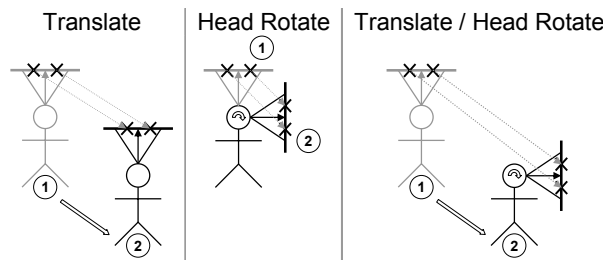


Figure 19 - AR working plane attached to the head can create vertices relative to the user but fixed to the world

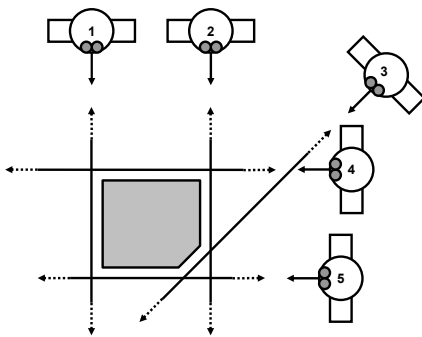


Figure 20 - Infinite planes created along head direction to quickly form a solid object

ing plane may also move and points may be created in world coordinates against the current surface. While this technique may be used to create complex collections of vertices, this can be tedious for many objects. Given a building with n walls (along with a roof and a floor), then $2n$ vertices need to be specified, and then connected together with $3n$ edges and $n+2$ facets.

To simplify the construction of many outdoor objects such as buildings, our construction at a distance concept includes the previously described infinite planes technique. The user specifies planes in the environment by sighting along the walls of a building and then projecting an infinite plane in this direction. This projection of planes is performed using the same techniques as depicted in Figure 8. The system can then compute a final solid based on the planes that have been specified, as depicted in Figure 20. Using the infinite planes, carving, and surface of revolution techniques outlined previously [23] [24], complex shapes can be defined more easily than specifying each vertex manually.

9 Conclusion

This paper has described the AR working planes technique and how it can be applied to performing 3D modelling within outdoor augmented reality. When working indoors, operations tend to occur within arms reach of the user, which allows direct manipulation techniques. In outdoor environments, objects tend to be at scales beyond that of the user, and at distances which are difficult to perceive accurately. When performing 3D modelling outdoors, new techniques are required to assist the user and AR working planes is one possible way to do this. Our techniques are based on working planes which are used in CAD systems, but since the mobile AR user can specify their viewpoint easily by just positioning their body, the specification of AR working planes is much more intuitive. The AR working planes proposed in this paper is a generic set of techniques that can perform any desired operation, so there is no need to supplement it with others. However, by adding other specialised techniques, the time needed for some modelling tasks can be reduced.

We have currently implemented a subset of these techniques in our Tinmith-Metro modelling application [23] [24]. While some of these techniques have been presented previously, this paper has presented a framework that unifies much of our existing work. This framework allows us to explore new combinations that were previously unthought of. The current Tinmith-Metro user interface implements various combinations of the techniques outlined, but does not provide the generic ability for the user to do any combination desired. We need to design a more generic user interface that allows the user to experiment with all possible combinations to discover which techniques work the best. We also need to perform usability studies to analyse which of the described techniques perform certain operations more efficiently.

10 References

- [1] Baillot, Y., Brown, D., and Julier, S. Authoring of Physical Models Using Mobile Computers. In *5th Int'l Symposium on Wearable Computers*, pp 39-46, Zurich, Switzerland, Oct 2001.
- [2] Billinghurst, M., Bowskill, J., Jessop, M., and Morphet, J. A Wearable Spatial Conferencing Space. In *2nd Int'l Symposium on Wearable Computers*, pp 76-83, Pittsburg, Pa, Oct 1998.
- [3] Bowman, D. *Conceptual Design Space - Beyond Walk-Through to Immersive Design*. In *Designing Digital Space*, New York, John Wiley & Sons, 1996.
- [4] Butterworth, J., Davidson, A., Hench, S., and Olano, T. M. 3DM: A Three Dimensional Modeler Using a Head Mounted Display. In *Symposium on Interactive 3D Graphics*, pp 135-138, Cambridge, Ma, Mar 1992.
- [5] Clark, J. H. *Designing Surfaces in 3-D*. Communications of the ACM, Vol. 19, No. 8, pp 454-460, 1976.
- [6] Cutting, J. E. and Vishton, P. M. *Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth*. In *Handbook of perception and cognition*, San Diego, Ca, Academic Press, 1995.
- [7] Cutting, J. E. *How the Eye Measures Reality and Virtual Reality*. Behaviour Research Methods, Instrumentation, and Computers, Vol. 29, pp 29-36, 1997.
- [8] Cutting, J. E. *Reconceiving Perceptual Space*. In *Perceiving Pictures: An Interdisciplinary Approach to Pictorial Space*, Cambridge, Ma, MIT Press, 2002.
- [9] Drascic, D. and Milgram, P. Perceptual Issues in Augmented Reality. In *SPIE Volume 2653: Stereoscopic Displays and Virtual Reality Systems III*, pp 123-134, San Jose, Ca, Jan 1996.
- [10] Feiner, S., MacIntyre, B., Haupt, M., and Solomon, E. Windows on the World: 2D Windows for 3D Augmented Reality. In *6th Annual Symposium on User Interface Software and Technology*, pp 145-155, Atlanta, Ga, Nov 1993.
- [11] Feiner, S., MacIntyre, B., and Hollerer, T. A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment. In *1st Int'l Symposium on Wearable Computers*, pp 74-81, Cambridge, Ma, Oct 1997.
- [12] Forsberg, A., Herndon, K. P., and Zeleznik, R. Aperture Based Selection for Immersive Virtual Environments. In *9th Annual Symposium on User Interface Software and Technology*, pp 95-96, Seattle, Wa, Nov 1996.
- [13] Hinckley, K., Pausch, R., Goble, J. C., and Kassell, N. F. A Survey of Design Issues in Spatial Input. In *7th Annual Symposium on User Interface Software and Technology*, pp 213-222, Marina del Rey, Ca, Nov 1994.
- [14] Julier, S., Lanzagorta, M., Baillot, Y., Rosenblum, L., Feiner, S., and Hollerer, T. Information Filtering for Mobile Augmented Reality. In *3rd Int'l Symposium on Augmented Reality*, pp 1-10, Munich, Germany, Oct 2000.
- [15] Kalkusch, M., Lidy, T., Knapp, M., Reitmayr, G., Kaufmann, H., and Schmalstieg, D. Structured Visual Markers for Indoor Pathfinding. In *1st Int'l Augmented Reality Toolkit Workshop*, Darmstadt, Germany, Sep 2002.
- [16] Liang, J. and Green, M. Geometric Modelling Using Six Degrees of Freedom Input Devices. In *3rd Int'l Conference on CAD and Computer Graphics*, pp 217-222, Beijing, China, Aug 1993.
- [17] Milgram, P. and Kishino, F. *A Taxonomy of Mixed Reality Visual Displays*. IEICE Trans. Information Systems, Vol. E77-D, No. 12, pp 1321-1329, 1994.
- [18] Mine, M., Brooks, F. P., and Sequin, C. H. Moving Objects In Space: Exploiting Proprioception In Virtual-Environment Interaction. In *Int'l Conference on Computer Graphics and Interactive Techniques*, pp 19-26, Los Angeles, Ca, Aug 1997.
- [19] Mine, M. R. *Virtual Environment Interaction Techniques*. Technical Report, University of North Carolina, Chapel Hill - Department of Computer Science, Report No. TR95-018, May 1995.
- [20] Mine, M. R. *Exploiting Proprioception in Virtual-Environment Interaction*. PhD Thesis, Department of Computer Science, University of North Carolina, Chapel Hill, NC, 1997.
- [21] Multigen. *SmartScene*. <http://www.multigen.com>
- [22] Piekarski, W., Gunther, B., and Thomas, B. Integrating Virtual and Augmented Realities in an Outdoor Application. In *2nd Int'l Workshop on Augmented Reality*, pp 45-54, San Francisco, Ca, Oct 1999.
- [23] Piekarski, W. and Thomas, B. H. Timmith-Metro: New Outdoor Techniques for Creating City Models with an Augmented Reality Wearable Computer. In *5th Int'l Symposium on Wearable Computers*, pp 31-38, Zurich, Switzerland, Oct 2001.
- [24] Piekarski, W. and Thomas, B. H. Interactive Augmented Reality Techniques for Construction at a Distance of 3D Geometry. In *7th Int'l Workshop on Immersive Projection Technology / 9th Eurographics Workshop on Virtual Environments*, Zurich, Switzerland, May 2003.
- [25] Pierce, J. S., Forsberg, A., Conway, M. J., Hong, S., Zeleznik, R., and Mine, M. R. Image Plane Interaction Techniques in 3D Immersive Environments. In *Symposium on Interactive 3D Graphics*, pp 39-43, Providence, RI, 1997.
- [26] Poupyrev, I., Billinghurst, M., Weghorst, S., and Ichikawa, T. The Go-Go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR. In *9th Annual Symposium on User Interface Software and Technology*, pp 79-80, Seattle, WA, Nov 1996.
- [27] Reitmayr, G. and Schmalstieg, D. Mobile Collaborative Augmented Reality. In *Int'l Symposium on Augmented Reality*, pp 114-123, New York, NY, Oct 2001.
- [28] Robinett, W. and Holloway, R. Implementation of Flying, Scaling, and Grabbing in Virtual Worlds. In *Symposium on Interactive 3D Graphics*, pp 189-192, Cambridge, Ma, Mar 1992.
- [29] Sachs, E., Roberts, A., and Stoops, D. *3-Draw: A Tool For Designing 3D Shapes*. IEEE Computer Graphics and Applications, Vol. 11, No. 6, pp 18-24, 1991.
- [30] Stoakley, R., Conway, M. J., and Pausch, R. Virtual Reality on a WIM: Interactive Worlds in Miniature. In *Conference on Human Factors in Computing Systems*, pp 265-272, Denver, Co, May 1995.
- [31] Suomela, R. and Lehtikoinen, J. Context Compass. In *4th Int'l Symposium on Wearable Computers*, pp 147-154, Atlanta, Ga, Oct 2000.
- [32] Sutherland, I. Sketchpad: A man-machine graphical communication system. In *IFIPS Spring Joint Computer Conference*, pp 329-346, Detroit, Mi, May 1963.
- [33] Ware, C. and Jessome, D. R. *Using the Bat: A Six-Dimensional Mouse for Object Placement*. IEEE Computer Graphics and Applications, Vol. 8, No. 6, pp 65-70, 1988.