(This is a sample cover image for this issue. The actual cover is not yet available at this time.)

Interacting with Computers 24 (2012) 10–24

Contents lists available at SciVerse ScienceDirect

Interacting with Computers

journal homepage: www.elsevier.com/locate/intcom

An interactive 3D movement path manipulation method in an augmented reality environment

Taejin Ha [a], Mark Billinghurst [b], Woontack Woo [a,*]

[a] GIST U-VR Lab, Gwangju 500-712, South Korea
[b] The HIT Lab NZ, University of Canterbury, Private Bag 4800, Christchurch 8140, New Zealand

ARTICLE INFO

Article history:
Received 25 October 2010
Received in revised form 11 April 2011
Accepted 29 June 2011
Available online 24 October 2011

Keywords:
Immersive augmented reality
Augmented reality authoring
Movement path editing
Tangible user interface
3D object selection and manipulation

ABSTRACT

In this paper, we evaluate a path editing method using a tangible user interface to generate and manipulate the movement path of a 3D object in an Augmented Reality (AR) scene. To generate the movement path, each translation point of a real 3D manipulation prop is examined to determine which point should be used as a control point for the path. Interpolation using splines is then used to reconstruct the path with a smooth line. A dynamic score-based selection method is also used to effectively select small and dense control points of the path. In an experimental evaluation, our method took the same time and generated a similar amount of errors as a more traditional approach, however the number of control points needed was significantly reduced. For control manipulation, the task completion time was quicker and there was less hand movement needed. Our method can be applied to drawing or curve editing methods in AR educational, gaming, and simulation applications.

1. Introduction

Augmented Reality (AR) that overlays virtual imagery on the real world is now becoming commonplace and many laboratories and companies are developing AR authoring methods and tools for easy creation of AR applications. However, most previous authoring work has been focused on registration and arrangement of 3D objects. To enhance the user interactions with augmented 3D objects, we need other authoring methods. In this paper, we consider movement path generation and manipulation methods for 3D objects in an AR scene. This is applicable to interactive and dynamic AR applications in education, gaming, design, animation, simulation, and many other areas where the movement path of a 3D object needs to be specified.

Generally, path manipulation methods in an AR environment are classified into three types: (i) use of third-party commercial modeling software, in which objects are made and positioned by 3D tools such as 3D Studio Max[1] or MAYA,[2] then loaded using an AR library/toolkit, (ii) extending commercial software to include an AR authoring feature in which an AR plug-in module is connected to conventional commercial software such as Director[3] (MacIntyre et al., 2004), XNA Game Studio[4] (Oda et al., 2008), Virtools[5] (Geiger et al., 2007), or Sketch up,[6] and (iii) use of a tangible user interface (TUI)-based immersive AR authoring method, in which a user can intuitively set a movement path using hand movements and immediately confirm the results in the same AR environment (Lee et al., 2004; Ha et al., 2010). Most of the previous work has been one of the first two types, but our research is focusing on the use of TUI methods.

Using commercial software requires professional training and the ability to repeatedly switch from a desktop graphics environment to the AR environment to check the final results. In contrast, the TUI-based immersive movement path generation method has the advantage that the 3D object path can be intuitively generated by direct use of hand movements. However, the generated path can have temporal errors introduced by the trembling of the user's hand and errors in AR tracking. Path saving and modification is also difficult because many small control points are saved and it can be difficult to select them in the AR environment; in this paper we address those issues.

We propose an immersive AR-based path manipulation method that combines the advantages of earlier methods with enhanced path editing functions. Specifically, by exploiting a virtual hand

* Corresponding author. Tel.: +82 629703157; fax: +82 629702204.
E-mail addresses: tha@gist.ac.kr (T. Ha), mark.billinghurst@hitlabnz.org (M. Billinghurst), wwoo@gist.ac.kr (W. Woo).

[1] 3D Studio Max Product Information http://usa.autodesk.com/3ds-max (2010.9.1).
[2] MAYA Product Information http://usa.autodesk.com/maya (2010.9.1).
[3] Adobe Director, http://www.adobe.com/products/director (2010.9.1).
[4] Microsoft XNA Game Studio, http://www.xna.com (2010.9.1).
[5] Virtools, www.virtools.com (2010.9.1).
[6] AR-media™ Plugin for Google™ SketchUp™, http://www.inglobetechnologies.com (2010.9.1).

0953-5438/$ - see front matter © 2011 British Informatics Society Limited. All rights reserved.
doi:10.1016/j.intcom.2011.06.006

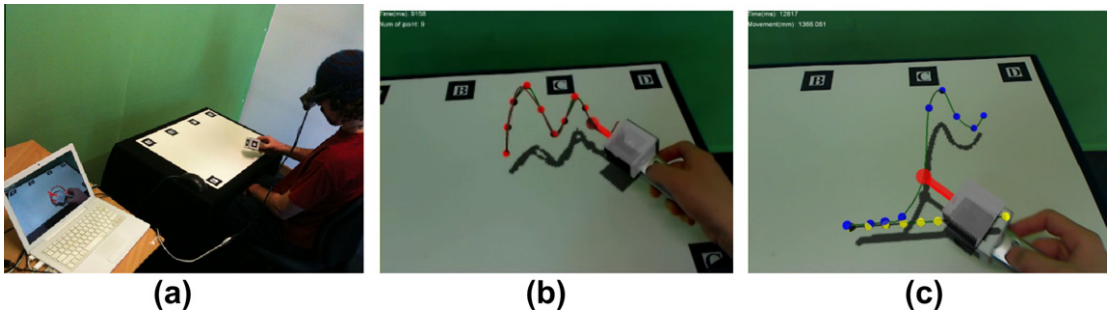

Fig. 1. Our AR system that uses a bi-ocular video see-through HMD and computer vision-based fiducial tracking: a user can (a) see an AR scene through the video see-through HMD with a camera aimed in the user's view direction; (b) generate a movement path using hand movements (control point in red) by using a 6 DOF manipulation prop; and (c) select a control point (blue point) and translate to the goal control point (yellow point).

technique based on a real 6 DOF (degree of freedom) manipulation prop, a user can directly and intuitively generate and manipulate the movement path of a 3D object, as shown in Fig. 1. To do this, we developed an allocation test method for movement position of the manipulation prop to generate proper control points. A reconstruction method then interpolates the movement path by using the saved control points and the spline curve model. In addition, a dynamic selection method effectively selects the small dense control points of the movement path based on distance, direction, frequency, and time score. Our research is the first to implement this method and the first to provide a rigorous user evaluation of TUI path manipulation methods in AR.

This paper is organized as follows. In Section 2, we examine the related work; in Section 3, we propose methods to supplement the related work. In Section 4, we describe our implementation and experimental results. Finally, our conclusions and futures are contained in Section 5.

2. Related work

2.1. 3D movement path generation in an immersive AR environment

Many studies have been conducted related to real-time 3D path generation in the Virtual Reality (VR) research field. The conventional 3D path-generation method has been adapted from desktop tools and uses 2D user interfaces within orthographic or perspective projections of a VR scene. This requires extended viewpoints to deal with a 3D movement path in all three coordinates and can be very inconvenient (Clark, 1976; Igarashi et al., 1999).

An enhanced approach can be realized in fully immersive virtual environments (IVEs) (Krüger et al., 1995; Buxton et al., 2000), particularly projection-based systems (PBVEs) (Wesche and Droske, 2000; Wesche and Seidel, 2001). In the IVEs, users experience virtual environment through single/multi large display devices. Interaction with the environment is through a 3D input device such as a space ball, a flying mouse called a BAT (Sachs et al., 1991; Liang and Green, 1994), data gloves (Shaw and Green, 1997; Schkolne et al., 2001), a haptic device (e.g., Phantom), and a 3D stereoscopic viewing system. Typically, an immersive VR system with 3D or 6 degree of freedom input device guarantees real-time and highly accurate path generation but requires external and high-cost tracking devices/installations (e.g., magnetic, infrared, or ultrasonic tracker) to track the user interface. In particular, the dis-collocated 3D control and fixed visual space may cause an extra cognitive load, as shown in Fig. 2a.

However, that has not been well studied in AR situations, in conventional AR authoring, the paths for moving 3D objects are generally created using modeling software within orthographic

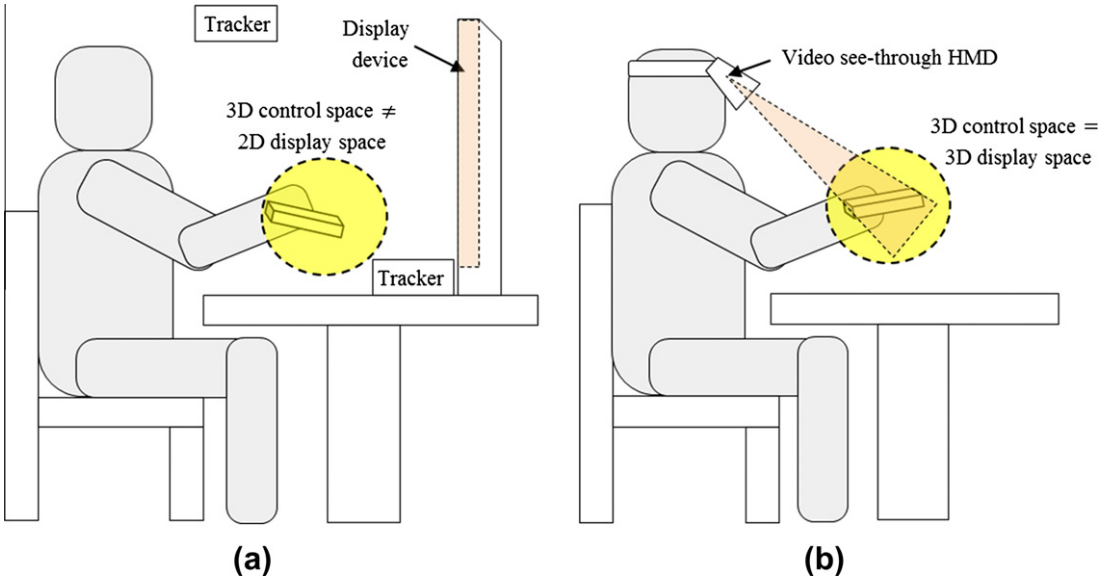

Fig. 2. (a) Immersive virtual environments (VEs): typically, an immersive VR system with 3D user input device guarantees real-time and highly accurate path generation but requires external and high-cost tracking devices/installations. In particular, the dis-collocated 3D control and fixed visual space may cause an extra cognitive load and (b) immersive AR environments: using an egocentric view with HMD and a real camera image and computer vision tracking of a tangible manipulation prop, along with graphic visualization corresponding to the user's viewpoint/direction, a user can directly generate the movement path of a 3D object in collocated 3D control and 3D visual space. Also, compared to VR tracking methods, its cost is much lower and it is easier for an average user to use.

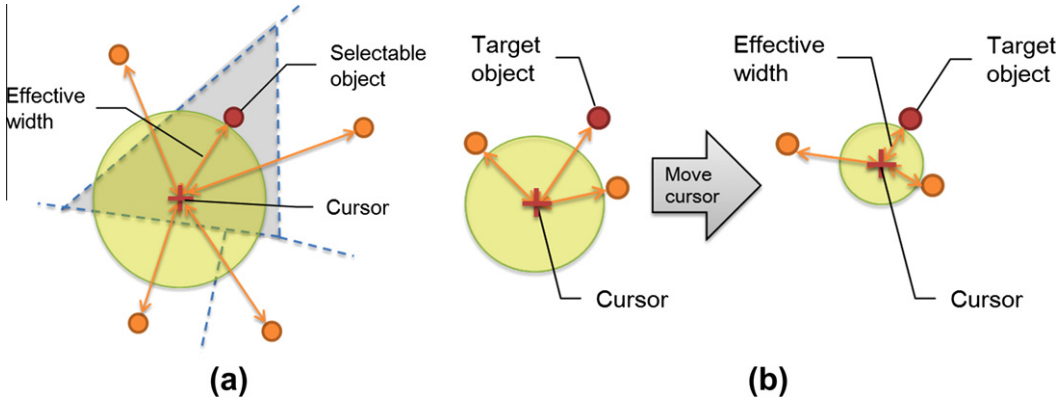

Fig. 3. 3D object selection method using the bubble cursor: (a) using the Voronoi diagram area and (b) a situation that can occur in a dense 3D object arrangement.

or perspective projections, and then the objects are viewed in AR using an AR library/toolkit or AR plug-ins[7,8]. If any modification of the movement path is required, a user must switch back to the modeling software, and then modify the movement path using a 2D user interface such as a keyboard, mouse, or tablet. Finally, an effort must be made to check the final results in an AR environment. This is similar approach with the conventional path generation method of VR environments and can be also tedious.

Therefore researchers have begun to start exploring immersive AR-based authoring methods, using an egocentric view with HMD and a real camera image and computer vision tracking of a tangible manipulation prop with fiducial tracking markers attached to it (Lee et al., 2004; Ha et al., 2010), as shown Fig. 2b. In this system, by using a bi-ocular video see-through HMD with a camera enables real-time head (HMD) tracking, graphic visualization corresponding to the user's viewpoint/direction, and a handheld 3D manipulation prop, a user can directly generate the movement path of a 3D object in collocated 3D control and 3D visual space, so immersive AR-based authoring methods can enhance spatial intuitiveness. Also compared to VR tracking methods, its cost is much lower and it is easier for an average user to use. Although immersive AR authoring systems have been demonstrated there has been no previous research on path manipulation in immersive AR applications using TUI tools, and no formal user evaluations presented.

Considering movement path generation, there are a number of methods that could be used, and the modeling method should appropriately reflect the external characteristics of a movement path and intuitively generated the path if needed with minimum inputs. The simplest approach is a polynomial interpolation. Namely, a movement path with n control points can be expressed using a polynomial equation with $n-1$ degrees of freedom; however, using many control points can require a much higher degree polynomial equation; therefore, the computational complexity is increased and real-time manipulation is difficult. In addition, erroneous undulations can occur.

To overcome the problem, spline interpolation can be exploited in which the movement path can be generated by adopting piecewise polynomial equations with 3 degrees of freedom less than usual. A spline interpolation is divided into two methods depending on how the curve passes through the control points; an approximating spline or an interpolating spline (Shirley et al., 2005). An approximating spline does not always pass through all the control points. To modify the shape of the curve a user must adjust the control points, tangent vector, knot, weight values, and other factors. Control of these kinds of parameters might not be easy and requires a lot of time (Bartels et al., 1987). On the other hand, an interpolating spline passes through a given control point, and the user can manipulate the control points to modify the curve. Thus providing an intuitive way for users to manipulate the control points is key for interactive path manipulation.

2.2. 3D control point selection for path manipulation

The control points of a saved path can be represented as sequential small, dense 3D objects. We need a method to efficiently select these objects in the AR environment. In this case, an egocentric 3D object selection method is normally used, such as ray casting or virtual hand methods (Bowman et al., 2004). With ray casting, a 3D object is selected by using an infinite virtual line from the user's hand. In the virtual hand method, a 3D object is selected if the user touches it with a virtual hand matching the user's physical hand position.

With ray casting a 3D object far away can be easily selected, but small changes of angle at the origin can cause a large displacement at the tip of the ray. So ray casting is not a good for selecting a very small 3D object at arms-length. To solve this problem, the following volume-based selection techniques have been developed: Spotlight (Liang and Green, 1994), Aperture (Forsberg et al., 1996), Image-plane (Pierce et al., 1997), and Depth Ray (Grossman and Balakrishnan, 2006).

Intuitive 6 DOF manipulations of a 3D object is another challenge (Bowman and Hodges, 1997). Virtual hand techniques are usually used in AR environments for 6 DOF manipulation of a 3D object (Kato et al., 2000; Kato et al., 2003; Lee et al., 2004; Ha et al., 2010). In particular, a mixed method using a tangible user interface (TUI) (Ishii and Ullmer, 1997; Billinghurst et al., 2001) provides an affordance through its physical form factor that can help a user intuitively determine a manipulation function and how to manipulate the 3D object.

The pointing-based selection method is one of the most popular virtual hand techniques. This involves checking distances between the position of a virtual hand and positions of 3D objects, and selecting an object within a specific area; however, this requires the user to perform an accurate collision with a 3D object in AR space with exact 3D depth perception, which may be difficult and tiring.

To solve these drawbacks, a volume-based selection method can be used. According to Fitts' law (Fitts, 1964), a large selection area (effective width) can lower the index of difficulty of the selection task (Vanacken et al., 2009); however, if the size of the selected area is very large, multiple objects can be selected simultaneously, and if their size is very small, volume-based selection

[7] AR-media™ Plugin for Google™ SketchUp™, http://www.inglobetechnologies.com (2010.9.1).
[8] Linceo VR, http://www.seac02.it (2010.9.1).

methods can lose their effectiveness. Therefore, a bubble cursor method, using a dynamically changing selection area, can be one possible choice (Grossman and Balakrishnan, 2005). In this case, considering the Voronoi diagram area, a 3D object with shortest distance to the cursor is automatically selected, as shown in Fig. 3a. The size of the semitransparent sphere is also changed dynamically to provide visual feedback. Therefore, a proper 3D object can be selected without exact positioning of the cursor in the 3D object.

However, in the case of dense objects (e.g., small control points), as shown in Fig. 3b, the cursor should be very close to an object position because the effective width for selection is smaller. This might reduce the advantage of a bubble cursor. In addition, frequent position fluctuations of the cursor in the air caused by a user's unstable hand movements can hinder the selection of a 3D object.

As an alternative, Haan suggested a 3D object selection model using a temporal relevance profile based on the distance between virtual objects on a table-top projection in the VR environment. Haan also implemented virtual ray bending to an object with maximum score as a visual effect and showed enhanced selection speed compared to conventional volume selection methods (Haan et al., 2005).

Steed extended Haan's work and suggested TVSFs (time-varying scalar fields) (Steed, 2006). This is a unified expression model that covers various conventional selection techniques. It comprises the following four steps: (1) a determination step for instantaneous relevance, (2) a computation step for total summation of relevance, (3) an object highlighting step based on the computation, and (4) an object selection step from among a set of objects using the total summation score. However, no concise implementation or verification by user experiment was presented.

Finally, Olwal designed a speech and gesture-based multimodal AR system and suggested a statistical and geometric algorithm to recognize an augmented object (Olwal et al., 2003). It dynamically selects a 3D object based on a volume area attached to a user's body, and five kinds of score (e.g., time, distance, stability, object visibility by pixel color and proximity priority); however, no experimental evaluation or analysis of the suggested algorithm was mentioned.

3. Path modeling and control point selection method

Fig. 4 shows the overall procedure for movement path generation and manipulation that we use in our AR environment. In the initial movement path generation step, as long as the user presses and holds a button on the manipulation prop, control points are generated through the control point allocation test, and a movement path is reconstructed as a smooth form in real-time. When the user releases the button, the path generation is completed. In the movement path manipulation process, a control point with the highest score can be selected and translated to a position where a user wants to move it; the movement path of the 3D object can then be updated in real-time.

3.1. Movement path generation

For tangible input we use a cube with multiple markers printed on it attached to a mouse pen input device (Ha and Woo, 2010) (see Fig. 1). By using the mouse input buttons, a discrete input generates the event functions necessary for an authoring interface. Multiple markers enable tracking of the tangible prop at arbitrary camera viewing angles and provide continuous input for manipulating the position/rotation matrixes of a 3D object.

A coordinate of a control point is first acquired from the transformation matrix $\mathbf{M}_{\text{Tip}}$ of a virtual line tip point, augmented from the front of the manipulation prop based on the world coordinate system $\mathbf{M}_{\text{World}}$, as shown in Fig. 5, where $\mathbf{M}_{\text{Tip}}$ is translated a

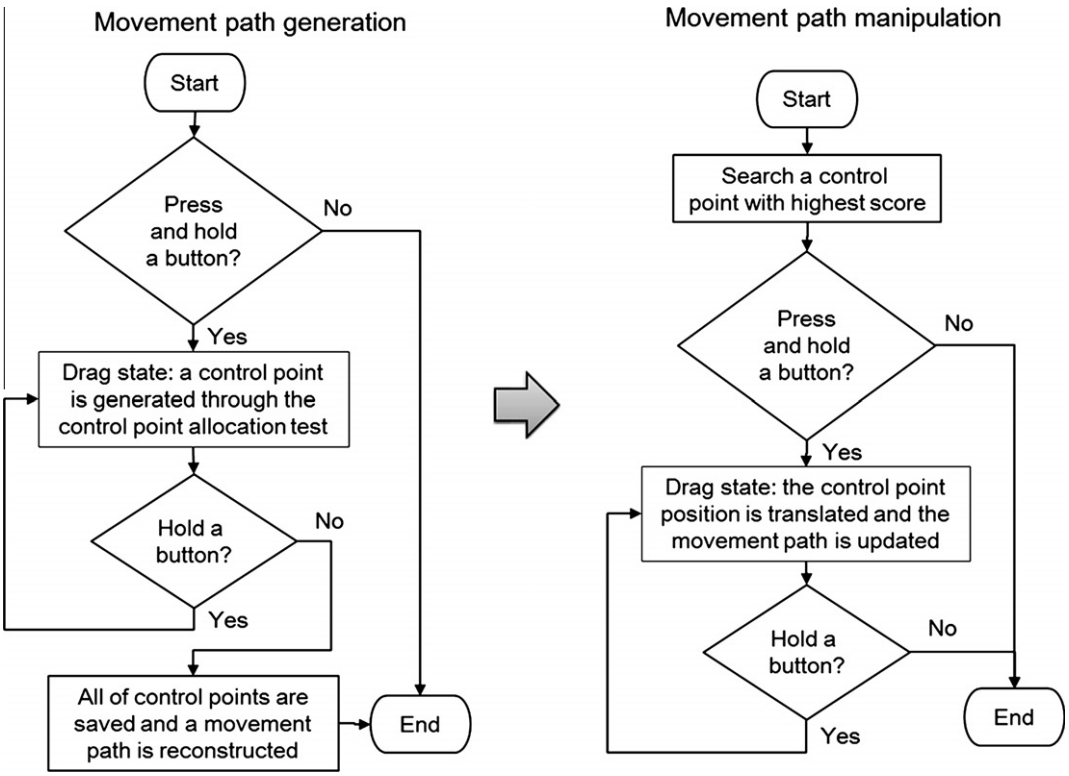

Fig. 4. Overall procedure for movement path generation and manipulation.

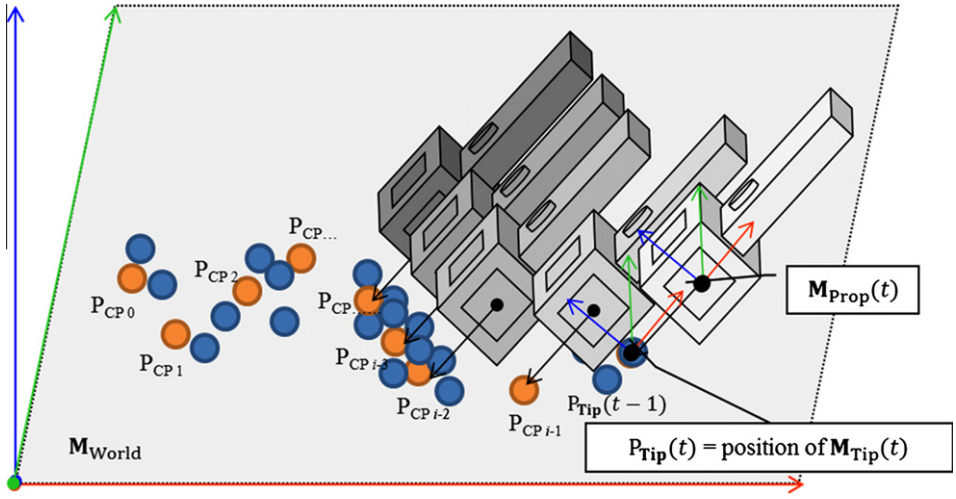

Fig. 5. The control point allocation method of a movement path: allocate a control point, considering (i) the distance between a previously saved control point and the manipulation prop and (ii) the dwell time of the manipulation prop around a specific position.

specific distance from the manipulation prop transformation matrix $\mathbf{M}_{\text{Prop}}$.

In conventional methods, the control points of a path are allocated whenever the manipulation prop is detected; however, the control points could be dense if there are only small movements of the manipulation prop. To solve this problem the control points are allocated over a specific interval between $P_{\text{CP } i-1}$, the position of the previously saved control point, and $P_{\text{Tip}}(t)$, the position of the current transformation matrix $\mathbf{M}_{\text{Tip}}(t)$, as shown in Eq. (1); therefore, the problem of dense control points is avoided.

To explore the details of the movement path, a user can slowly move the manipulation prop around a specific line. A control point can also be created if the position of the manipulation prop continuously dwells at a specific distance, as shown in Eqs. (2) and (3).

Using Eq. (3), if $D_{\text{CP\_Tip}}$, the distance between a control point and the tip of manipulation prop, or T_{Dwell}, the dwell time of the manipulation prop at specific section, are each greater than a specific threshold, the corresponding coordinate of the tip of the manipulation prop $P_{\text{Tip}}(t)$ is allocated as control point $P_{\text{CP } i}$. This method saves storage space by not recording all the path points and the number of control point selections required to manipulate a path later.

$$D_{\text{CP\_Tip}} = |P_{\text{CP } i-1} - P_{\text{Tip}}(t)| \tag{1}$$

$$T_{\text{Dwell}} = \begin{cases} \sum 1, & \text{if } |P_{\text{Tip}}(t-1) - P_{\text{Tip}}(t)| < D_{\text{Thres}} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$P_{\text{CP } i} = P_{\text{Tip}}(t), \text{ if } D_{\text{Cp\_Tip}} > D_{\text{Thres}} \text{ or } T_{\text{Dwell}} > T_{\text{Thres}} \tag{3}$$

The movement path can be reconstructed using the control points and the parametric curve equation, so the path is converted to a smooth shape. This can compensate for the jittering effect caused by a user's hand trembling or unstable AR tracking. Among various interpolating splines, we use the Catmull–Rom spline (Catmull and Rom, 1974) which has less extension than other conventional splines, but the appearance of the curve can be easily manipulated with control points. In the Catmull–Rom spline, a curve segment $q(t)$ is defined by the four given control points CP_{i-1}, CP_i, CP_{i+1}, CP_{i+2}, and ratio variable t, using Eq. (4). Where t ($t \in [0,1]$) means distance ratio between the nearest two control points (e.g., from CP_i to CP_{i+1}).

$$P(t) = 1/2[1.0 \quad t \quad t^2 \quad t^3] \begin{bmatrix} 0 & 2 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 2 & -5 & 4 & -1 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} \text{CP}_{i-1} \\ \text{CP}_i \\ \text{CP}_{i+1} \\ \text{CP}_{i+2} \end{bmatrix} \tag{4}$$

3.2. Movement path manipulation

To overcome the disadvantages of 3D object selection methods such as exact position matching, we use a dynamic control point selection method using four score measures, as shown in Fig. 6. The first score is related to the distance from the center position of the selection volume, where a control point object at close distance has a high score. In Eq. (5), D_{Radius} is the radius length of the volume, and $D_{\text{CP } i}(t)$ is the distance from the center position of the volume to the control point object CP_i.

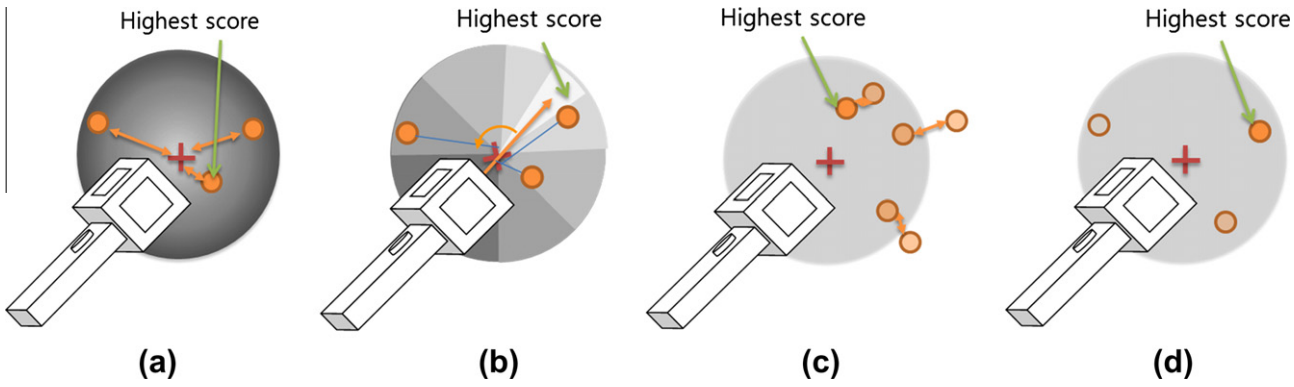

Fig. 6. The dynamic control point selection method using four scores: (a) distance score; (b) direction score; (c) frequency score; and (d) time score.

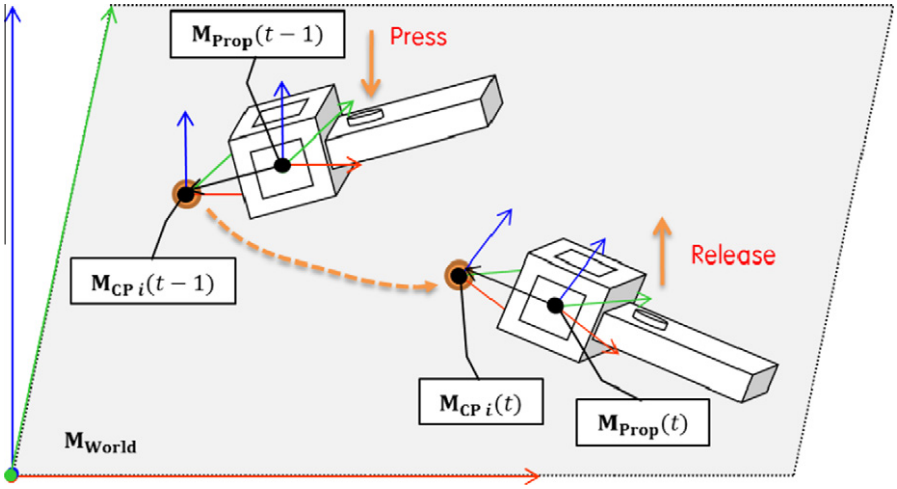

Fig. 7. Using the 6DOF manipulation prop, the position of a control point object can be changed and the movement path can be updated in real-time.

$$S_{\text{DisCP }i}(t) = 1 - D_{\text{CP }i}(t)/D_{\text{Radius}},\ \text{where } S_{\text{DisCP }i}(t) \in [0,1] \tag{5}$$

The direction score computes the direction angle of a control point object based on the direction of the manipulation prop when the control point object is within the manipulation prop's volume. This allows a user to select a control point in the intended direction. In Eq. (6), the angle (radian) of a control point object CP_i is obtained using the inner product of the direction vector V_{prop} of the manipulation prop and the direction vector $V_{\text{CP }i}(t)$ of a control point object CP_i based on the center position of the volume, where x ranges from -1 (opposite direction) to 1 (same direction). Using the relationship as defined in Eq. (7), $S_{\text{DirCP }i}(t)$ is adjusted from 0 to 1, with the score close to 1 if a control point is in the same direction, and 0 if a control point is in the opposite direction.

$$\text{Angle}_{\text{CP }i}(t) = \cos^{-1}\left(x = V_{\text{Prop}} \cdot V_{\text{CP }i}(t)/|V_{\text{Prop}}||V_{\text{CP }i}(t)|\right) \tag{6}$$

$$S_{\text{DisCP }i}(t) = 1 + (x-1)/2,\ \text{where } S_{\text{DisCP }i}(t) \in [0,1] \tag{7}$$

The frequency score reflects the number of times a control point object is included in the volume, with frequent inclusion leading to a high score. As shown in Eqs. (8) and (9), the frequency of a control point increases if the control point is contained in the volume; otherwise frequency decreases. Finally, in Eq. (10), the frequency $F_{\text{CP }i}(t)$ of a control point object CP_i at current frame t is normalized to a range of 0–1 by dividing it by F_{Upper}, where F_{Upper} is maximum frequency value among all control points.

$$\text{In}_{\text{CP }i}(t) = \begin{cases} 1, & \text{if } |P_{\text{CP }i}(t) - P_{\text{Prop}}(t)| < D_{\text{Radius}} \\ -1, & \text{otherwise} \end{cases} \tag{8}$$

$$F_{\text{CP }i}(t) = \sum \text{In}_{\text{CP }i}(t),\ \text{where } F_{\text{CP }i}(t) = 0 \text{ if } F_{\text{CP }i}(t) < 0 \tag{9}$$

$$S_{\text{FreqCP }i}(t) = F_{\text{CP }i}(t)/F_{\text{Upper}},\ \text{where } S_{\text{FreqCP }i}(t) \in [0,1] \tag{10}$$

There is a time score related to the successive time an object is contained in a volume. The longer a control point is continually contained in a volume, the higher its time score. However, the frequency is set to 0 if a control point is not contained in the volume (Eq. (11)). In Eq. (12), the range of $T_{\text{CP }i}(t)$ of control point object CP_i at frame t is modified from 0 to 1 by dividing by T_{Upper}, where T_{Upper} is maximum time value among all control points.

$$T_{\text{CP }i}(t) = \begin{cases} \sum 1, & \text{if } |P_{\text{CP }i}(t) - P_{\text{Prop}}(t)| < D_{\text{Radius}} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

$$S_{\text{TimeCP }i}(t) = T_{\text{CP }i}(t)/T_{\text{Upper}},\ \text{where } S_{\text{TimeCP }i}(t) \in [0,1] \tag{12}$$

The total summation score $S_{\text{Contrib CP }i}(t)$ of a control point object CP_i at current frame t is calculated from the four scores (see Eq. (13)), where α, β, γ, and ω are weighted values for each score and the sum of them is 1 ($\alpha, \beta, \gamma, \omega \in [0,1]$). At every frame, the current score value is summed with the previous frame's score value, using Eq. (14), where each C_s and C_g is a weight value, meaning a decreasing ratio of the total sum to the previous score and increasing ratio of the total sum to the current score, respectively. The sum of C_s and C_g is 1 ($C_s, C_g \in [0,1]$). Finally, using Eq. (15), a control point with the maximum score value at current time t is determined as the selected control point, CP_i.

$$S_{\text{Contrib CP }i}(t) = \alpha S_{\text{DisCP }i}(t) + \beta S_{\text{DisCP }i}(t) + \gamma S_{\text{FreqCP }i}(t) + \omega S_{\text{TimeCP }i}(t) \tag{13}$$

$$S_{\text{Total CP }i}(t) = C_s S_{\text{Total CP }i}(t-1) + C_g S_{\text{Contrib CP }i}(t) \tag{14}$$

$$\text{CP}_i = \arg_{\max i}\ S_{\text{Total CP}i}(t) \tag{15}$$

After the control point is selected, its position can be changed by dragging it using the manipulation prop (Fig. 7). During the drag state, the position property of control point CP_i's transformation matrix $\mathbf{M}_{\text{CP }i}$ is replaced by the virtual ray's transformation matrix $\mathbf{M}_{\text{Tip}}$ augmented in front of the manipulation prop (Ha and Woo, 2010). In this way the movement path of the 3D object is updated in real-time.

4. Experiment

4.1. Experiment design

We conducted two experiments to evaluate the performance of the movement path generation and manipulation methods. The first evaluates the process of movement path generation (Fig. 8a). A user moves the manipulation prop along the target movement path positioned in 3D space. At the same time, a movement path is generated. We compared our new Catmull–Rom spline-based method (SPLINE) with the conventional method, which saves the manipulation prop position at every frame (NO_SPLINE). The goal of this experiment is to explore if the SPLINE method can produce paths as accurate as the NO_SPLINE method.

In the second experiment, the task is to match as closely as possible the control points of a given movement path to the control points of the target movement path in 3D space (Fig. 8b). In this experiment, we evaluate how quickly and accurately a user selects

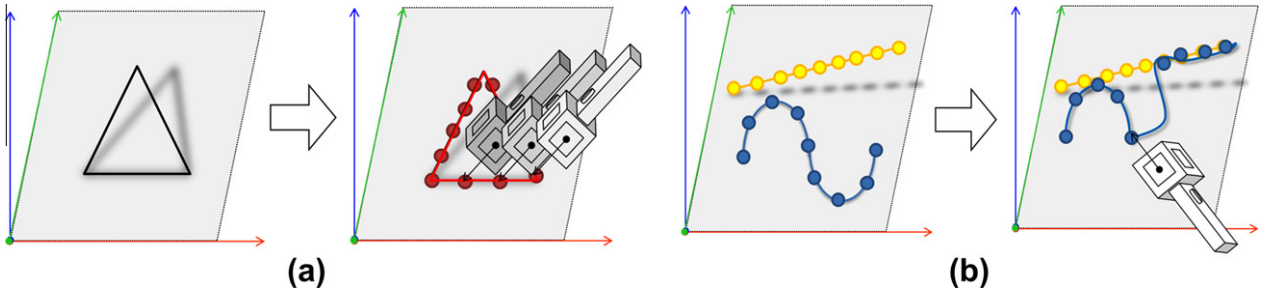

Fig. 8. (a) Experiment 1. Comparison of movement path generation methods: a target movement path is given, and a user moves the manipulation prop along the target movement path and (b) Experiment 2. Comparison of the control point selection method: there are control points in a target movement path and given control points a user can manipulate, and a user must select and move the given control points to the control points of the target movement path as accurately and quickly as possible.

Table 1
Independent variables.

Experiment	Independent variables
Common condition	Path shape: (i) equilateral triangle, (ii) square, (iii) circle, and (iv) free-form curve
Experiment 1: movement path generation	Method: (i) conventional movement path generation method (NO_SPLINE) and (ii) Catmull–Rom spline based method path generation method (SPLINE) Number of trials: total 8 trials (4 paths × 2 methods)
Experiment 2: control point selection	Method: (i) Point based selection method (PS: Point Selection), (ii) Voronoi volume based selection method (VS: Volume Selection), and (iii) Score based selection method (SS: Score Selection) Arrangement: (i) a goal path closer and (ii) further than a given path to a user Number of trials: total 12 trials (4 paths × 3 methods × 1 of 2 arrangements)

Table 2
Dependent variables.

Experiment	Dependent variables
Experiment 1: movement path generation method	Number of points Error (mm) Completion time (ms)
Experiment 2: control point selection method	Number of clicks Effective distance (mm) Movement distance of the manipulation prop (mm) Completion time (ms)
Subjective user ratings	Accuracy, speed, easiness, comfort, and understandability (Likert scales from 1 to 7)

and manipulates the control points. We compare our score measure-based selection method (SS: Score Selection) to a point-based selection method (PS: Point Selection), and Voronoi volume-based selection method of the bubble cursor (VS: Volume Selection). The goal of this experiment is to explore which selection method is the most accurate and efficient.

To conduct the above experiments, a within-subject repeated measures design was used. Table 1 shows the independent variables in the experiments. Movement paths in the common test condition are an equilateral triangle, square, circle, and free-form sine wave curve, which can be drawn with one movement of the manipulation prop. All of these are positioned in 3D space.

Experiment 1 examines four movement paths and two movement path modeling methods (total 8 trials), and Experiment 2 comprises four movement paths, three control point selection methods, and one of two arrangements to explore the effect of different arrangements between the target and given movement paths (e.g., a goal path closer or further than a given path to a user) (for a total of 12 trials). In both experiments, the first method is used to set the baseline result.

Table 2 shows the dependent variables in the experiments. In Experiment 1 these are the number of control points for movement path generation, the size of errors, and the time needed to complete the task. Errors are computed as the shortest distance between the generated control points and the target movement path. The completion time is measured from the time a user pushes a button of the manipulation prop until he or she releases it.

The dependent variables in Experiment 2 are the number of clicks, the effective distance for a selectable control point, the movement distance of the manipulation prop, and the completion time. The task is completed when a user moves each control point to within 10 mm of the corresponding control point of the target movement path. Otherwise, a user selects the control point and moves it within the error range again (increasing the number of clicks). The effective distance is the distance between a control point and the manipulation prop when the control point selection is selectable and a user pushes a button. The completion time is the time from the initial click to task completion. Any distance measure (e.g., error and distance) is calculated in the Euclidean 3D space.

In addition, user subjective ratings of the accuracy (How accurate was it to complete the task?), speed (How fast was it to complete the task?), easiness (How easy was it to complete the task?), comfort (How comfortable was it to complete the task?), and understandability (How understandable was it to complete the task?) were surveyed using a Likert scale with seven values, rated between 1 (most negative) and 7 (most positive).

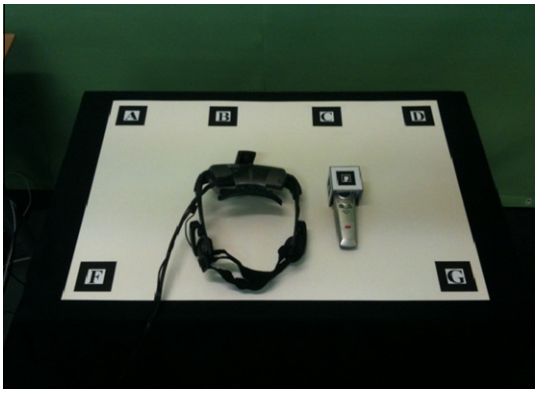

Fig. 9. The experimental environment: apparatus and the tracking pattern board (world coordinate system) in front of user on the table.

For statistical analysis of the results, we first ran Levene's test to assess the equality of variance. We then used an independent *t*-test to compare the two kinds of movement path generation methods in experiment 1 and for the comparison of the two kinds of movement path arrangements in experiment 2.

We also used a one-way ANOVA (Analysis of Variance) test to compare the three kinds of control point selection methods used in experiment 2. In the post hoc analysis, we performed the Scheffe adjustment for equal variance assumption and the T3 adjustment of Dunnett for unequal variance assumption.

To analyze the user subjective ratings with the Likert scales, we used the Mann–Whitney U test for the two kinds of movement path generation methods in experiment 1 and for comparing the control point selection methods in experiment 2, and the Kruskal–Wallis test for the three kinds of control point selection methods used in experiment 2.

4.2. Implementation

The experiment was conducted in an indoor environment with controlled lighting for reliable computer vision tracking (Fig. 9). A video see through HMD[9] was used with resolution of 800 × 600 pixels, a 100 Hz-refresh rate, 26 degrees diagonal field of view, and weighing 7 oz. A general purpose USB camera[10] was attached onto the HMD capturing 30 image frames per second with 640 × 480-pixel resolution, connected to a computer equipped with duo 2 GHz CPU and 4 GB memory. A commercial pen mouse,[11] 135 × 39.5 × 24.5 mm in size, was attached to the physical prop for button input. For stable marker tracking, the auto exposure function of the camera was enabled and the camera was focused on a tracking board. Camera calibration was also conducted using calib_camera2 program of the ARToolKit library.[12] The osgART library[13] (Looser et al., 2006) was used for graphic rendering through OpenSceneGraph library,[14] and for vision-based marker tracking through the ARToolKit library.

To enable reliable AR tracking multiple tracking patterns were used. The board was 300 × 600 mm so the tracking pattern was positioned in camera view at all times. The size of each marker in the board was 55 mm, and the markers on the manipulation prop were 50 mm, including a 7 mm margin on each side. A marker was attached to each side of the cube except on the side with the handle (total 5 sides).

Fig. 10 shows the target movement paths (e.g., equilateral triangle, square, circle, and free-form curve) used in the experiment 1, and Fig. 11 shows the target paths (yellow[15] straight line) and the given movement paths (e.g., blue equilateral triangle, square, circle, and free-form curve) with control points. The target paths are further than the given movement paths in arrangement 1, and vice versa in arrangement 2.

To provide visual feedback, the control points (e.g., generated control points in experiment 1, and a selectable control point in experiment 2) were a different color (e.g., red), all the virtual objects had a shadow to help the user more easily perceive spatial depth in the AR environment, and a sound effect was played whenever a button on the handle was pushed or released.

As mentioned in Section 4.1, the NO_SPLINE method saves the manipulation prop position whenever it is detected, whereas the SPLINE method considers the distance and dwell time of the control points and the manipulation prop. In the experiment, the parameter D_{Thres} was set to 30 mm and T_{Thres} to 20 frames. In other words, a control point could be allocated if the manipulation prop was held at the same position for more than 20 frames, or the distance between the last control point and the manipulation prop was more than 30 mm.

For experiment two, the movement path consisted of 10 control points. For the point-based selection method (PS), a control point close to a 10 mm distance range was selectable. In the Voronoi volume-based selection method (VS), the nearest control point was selectable, and in the score-based selection method (SS), a control point with optimized score was selectable. The D_{Radius} was set to 100 mm, each weighted value of score α, β, γ, and ω was set to 0.25, and each decreasing ratio of total sum to previous score C_s and increasing ratio of total sum to current score C_g value was set to 0.5.

4.3. Experiment results and discussions

4.3.1. Experiment 1. Movement path generation

A total of paid 15 participants, 14 male and 1 female students with an average age of about 22 participated in our evaluation. Of the 15 participants, 10 were majors in computer science/engineering. All participants had experience with 3D movies, 3D computer games or 3D software, were right-handed, had no neurological/psychiatric disorders, and did not suffer from motion sickness/vertigo.

Participants were given an explanation of the technology using video clips and a demonstration, and then they were given time to practice the techniques themselves for about 10 min. The real tests took about 10 min. We asked the participants to follow the target path as quickly and as accurately as possible.

4.3.1.1. Error amount. Table 3 and Fig. 12 show the results of experiment 1. In general, the average error in the SPLINE method was a little smaller than that of the NO_SPLINE method, but with no significant difference between them ($t_{(118)} = 0.355$, $p = 0.723$).

4.3.1.2. Completion time. With the SPLINE method, the completion time was similar or decreased by about 1 s compared to the NO_SPLINE, but there was also no significant difference between them ($t_{(118)} = 0.365$, $p = 0.716$).

4.3.1.3. Number of points. However the SPLINE method was able to generate movement paths using only 7.8% to 9.1% of the control

[9] HMD, i-glasses PC/SVGA, http://www.i-glassesstore.com (2010.9.1).
[10] Camera: Logitech Webcam C905, http://www.logitech.com (2010.9.1).
[11] Wireless presenter: 3M WP-8000, http://www.3m.com (2010.9.1).
[12] ARToolKit, http://www.hitl.washington.edu/artoolkit (2010.9.1).
[13] osgART, www.artoolworks.com/community/osgart (2010.9.1).
[14] OpenSceneGraph, http://www.openscenegraph.org/projects/osg (2010.9.1).

[15] For interpretation of color in Figs. 1–3, 5–11, 13, 16, and 17, the reader is referred to the web version of this article.

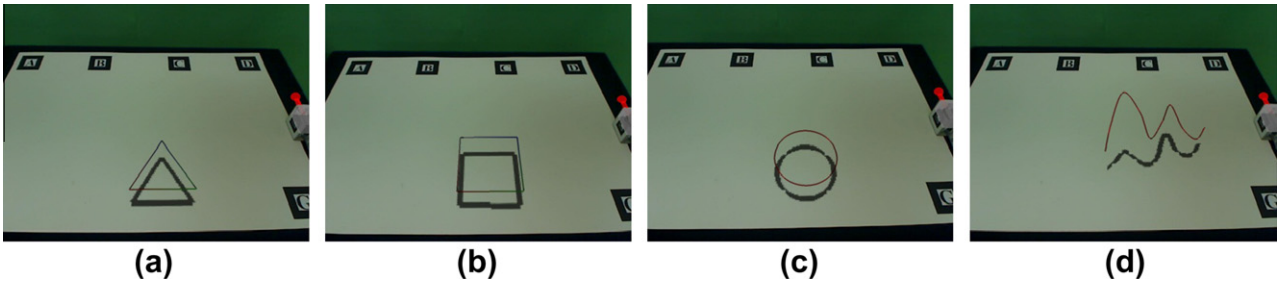

Fig. 10. The movement paths tested in the experiment 1: (a) equilateral triangle; (b) square; (c) circle; and (d) free-form curve.

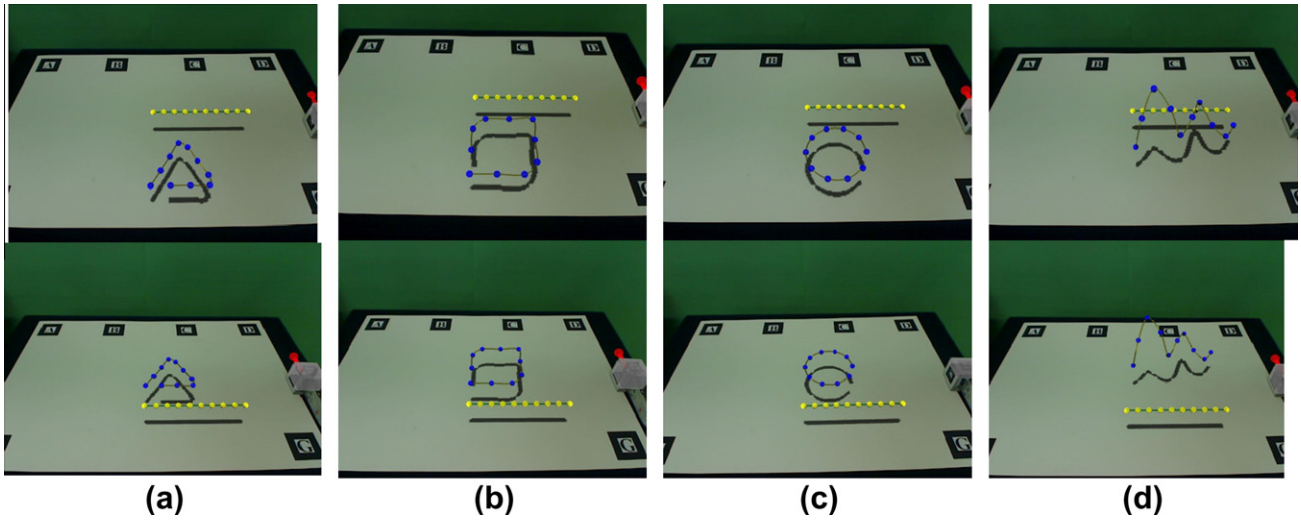

Fig. 11. The movement paths tested in the experiment 2, two kinds of arrangements for each path: (a) equilateral triangle; (b) square; (c) circle; and (d) free-form curve. The target paths (yellow control points) are further than the given movement paths (blue control points) in arrangement 1, and vice versa in arrangement 2.

Table 3
Descriptive statistics for the two kinds of the path generation methods (averages with standard deviations in parentheses).

	Equilateral triangle		Square		Circle		Free-form curve	
	NO_SPLINE	SPLINE	NO_SPLINE	SPLINE	NO_SPLINE	SPLINE	NO_SPLINE	SPLINE
Error (mm)	14.7 (±8.2)	17.2 (±9.6)	17.5 (±8.5)	14.3 (±5.2)	17.1 (±6.8)	16.0 (±6.6)	25.1 (±12.7)	24.4 (±13.2)
Completion time (ms)	13048.5 (±6415.9)	11927.8 (±4439.1)	16329.2 (±5924.3)	15952.5 (±6639.4)	13,243 (±6444.4)	13081.7 (±4984.3)	18517.6 (±7506.6)	18400.1 (±7921.5)
Number of points	200.9 (±53)	15.7 (±5.2)	229.1 (±43.5)	20.3 (±7.4)	200.3 (±55.0)	16.7 (±5.2)	244.5 (±51.3)	22.1 (±7.7)

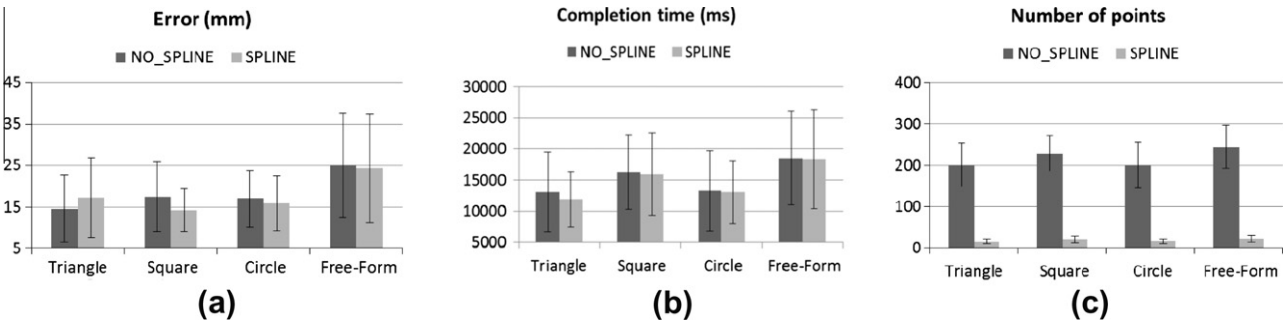

Fig. 12. Block charts for results of experiment 1: (a) error; (b) completion time; and (c) number of control points.

points used in the conventional NO_SPLINE method. There was significant difference between them ($t_{(60.965)} = 28.922$, $p < 0.001$).

Fig. 13 plots the cumulative distribution of control points. In the case of the NO_SPLINE method, many points (magenta color) were placed along the target movement path (black line). Using the SPLINE method, a relatively small number of control points were placed (blue points).

4.3.1.4. User subjective ratings. The SPLINE method showed significantly positive feedback for speed, and a slight positive preference for accuracy, easiness, and comfort, but a slight negative prefer-

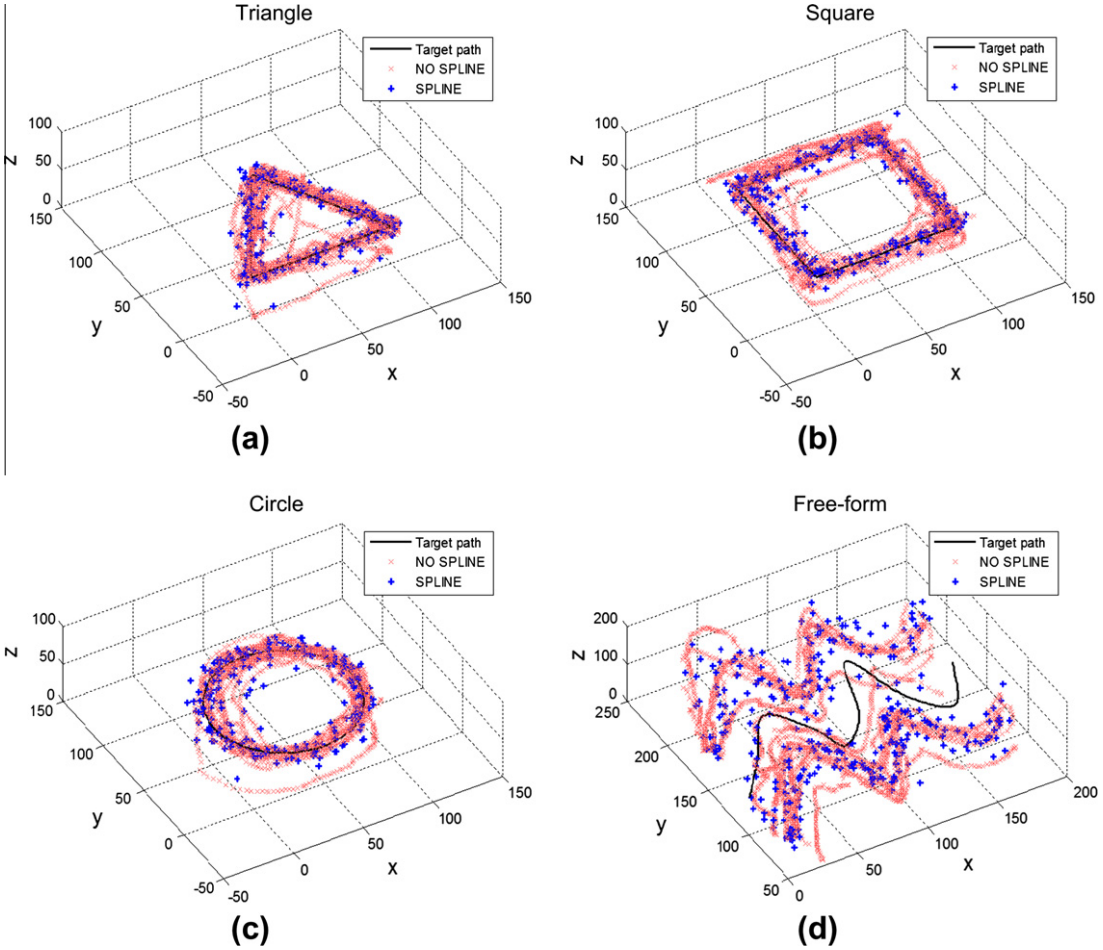

Fig. 13. Accumulated distribution charts of control points. The black line is the target curve, small magenta circles are allocated control points by NO_SPLINE, and blue circles are allocated control points by SPLINE method: (a) equilateral triangle; (b) square; (c) circle; and (d) free-form curve.

Table 4
User subjective ratings of experiment 1 using Likert scales (1–7 score; 1 being most negative and 7 being most positive, # users assigning the rank. A total of 15 participants participated in our evaluation).

Item	Method	1	2	3	4	5	6	7	Average
Accuracy	NO_SPLINE	0	1	2	0	8	4	0	4.8
	SPLINE	0	0	1	5	6	1	2	4.9
Speed	NO_SPLINE	0	0	2	7	6	0	0	4.3
	SPLINE	0	0	0	5	6	4	0	4.9
Easiness	NO_SPLINE	0	1	4	2	7	1	0	4.2
	SPLINE	0	0	2	5	7	1	0	4.5
Comfort	NO_SPLINE	0	2	3	4	5	1	0	4.0
	SPLINE	0	0	3	5	5	2	0	4.4
Understandability	NO_SPLINE	0	0	1	1	4	5	4	5.7
	SPLINE	0	0	2	0	5	4	4	5.5

ence for understandability, as shown in Table 4. Users felt that the SPLINE method was only significantly faster than the NO_SPLINE method ($Z = -2.090$, $p = 0.037$).

4.3.1.5. Findings. In the experiment, the error was relatively higher than previous work: generally, the marker tracking error was known as about 5–10 mm error within an arm's reach distance (i.e., about <600 mm), when an 80 mm-sized marker was used to measure accuracy (Kato and Billinghurst, 1999). Such an outcome may be attributed to the fact that the use, when wearing a HMD with a camera, freely moves his/her head and hand in 3D space while grasping a prop, while the camera simultaneously tracts the small fiducials of the base plane and manipulation prop. The error is an important factor that depends on its application. For applications such as a tangible AR-based education (Ha et al., 2011) that may not require high accuracy, our method would be applicable.

Considering the completion time, users felt that the SPLINE method was significantly faster than the NO_SPLINE method, but the time was not significantly different. This could be because of the different visual feedback provided: for NO_SPLINE the control points (red spheres) were sequentially generated following the

Table 5
Descriptive statistic for the arrangement and the path manipulation methods (averages with standard deviations in parentheses).

Arrangement	Method	Number of clicks	Effective width (mm)	Movement distance (mm)	Completion time (ms)
1	PS	12.5 (±3.6)	6.2 (±1.3)	3877.3 (±836.9)	64326.1 (±20243.2)
	VS	10.4 (±0.6)	31.1 (±7.9)	3330.7 (±754.0)	45838.5 (±14177.7)
	SS	10.6 (±1.3)	31.5 (±11.2)	3422.8 (±699.1)	45903.7 (±14890.7)
2	PS	13.4 (±4.2)	8.0 (±0.8)	4419.4 (±1170.7)	55447.7 (±13998.5)
	VS	10.6 (±0.8)	38.1 (±10.0)	3637.7 (±858.9)	37657.1 (±9667.2)
	SS	10.3 (±0.5)	41.3 (±11.2)	3259.4 (±548.1)	35307.9 (±8826.7)

prop movement (many points) but for SPLINE a curved line was generated only when a control point was allocated (a small number of points). For SPLINE, the completion time could have been even shorter if participants noticed that a curve could be connected between control points even with fast movement of the manipulation prop, but most participants missed this.

Our spline-based movement path generation method reduced the number of control points by more than 91% compared to a more conventional method, without any difference in error or task completion time. Our method saves only the control points and reconstructs a movement path, and provides the user with an opportunity to correct the movement path after it has been created. In addition, the reduced number of control points can drastically decrease the number of control point selections required for movement path modification.

4.3.1.6. Threshold values for the control point allocation method. In our implementation, the value of D_{Thres} in Eq. (3) was set to 30 mm. This means that a great deal of movement is needed before a control point is generated. If we decrease the value, then the generated path has an increased number of control points and a more detailed shape, but makes it more difficult to select the control points afterwards. However, if we increase the value, then we can expect the opposite effect. In future work, we can reduce the number of control points based on the speed, curvature and direction features (Han and Conti, 2005).

In Eq. (3), the value of T_{Thres} was set to 20 frames. This means that, if the prop dwells within distance D_{Thres} and over time T_{Thres}, then a control point is created. If it is lower than 20 frames, more detailed movement path control points could be made but with an increased number of points. In contrast, if the value is over 20 frames, we can obtain composite effects. On the other hand, the count of T_{Thres} can be changed depending on the camera frame rate, so we may need to manipulate system time rather than the frame count.

4.3.2. Experiment 2. Movement path manipulation

A total of 18 participants participated in our evaluation (arrangement 1: 8 male and 1 female, average age of about 23, 6 major in computer science/engineering; arrangement 2: 9 male, average age of about 22, 7 major in computer science/engineering). All participants in the first experiment were overlapped in the participant pools. All participants had experience with 3D movies, 3D computer games or 3D software experience, were right-handed, had no neurological/psychiatric disorders, and did not suffer from motion sickness/vertigo.

Participants were also given an explanation using video clips and demonstration of each technique for about 10 min and then they were able to practice the techniques for about 10 min. The user tests took about 20 min. We asked the participants to select given control points and to translate to the target control points as quickly as possible considering the characteristics of each method.

Table 5 and Fig. 14 show the experiment results from the two different arrangements, and Figs. 15 and 16 show the experiment result from the different arrangements and path types.

4.3.2.1. Number of clicks. In all paths, the PS showed significant difference with the VS and SS with 2–3 more clicks (post hoc analysis. $p < 0.001$), except on the free-form curve. On the other hand, there were no significant differences in number of clicks between the VS and SS conditions.

4.3.2.2. Effective distance. The effective distance of the PS method was significantly shorter (post hoc analysis. $p < 0.001$), but in cases where the VS and SS methods extended the selection area, the effective distance of possible selection was increased to a range of about 27–29 mm with a no difference between them in arrangement 2 (post hoc analysis. $p = 0.492$). This could mean that control point selection is possible by small movements of the manipulation prop. The initial selection position was much closer to the positions of the target control points in the PS method. In the VS and SS methods, the initial selection was relatively far from the target path, as shown in Fig. 16.

Considering the *t*-test results for arrangements and path types, the SS in square ($t_{(10.882)} = -2.182$, $p = 0.052$) and free-form curve ($t_{(16)} = -3.106$, $p = 0.007$), and the VS in square ($t_{(16)} = -8.317$, $p < 0.001$) showed significant increased effective distances in arrangement 2.

4.3.2.3. Movement distance. The PS method required the longest movement distance. The VS method needed an almost significantly

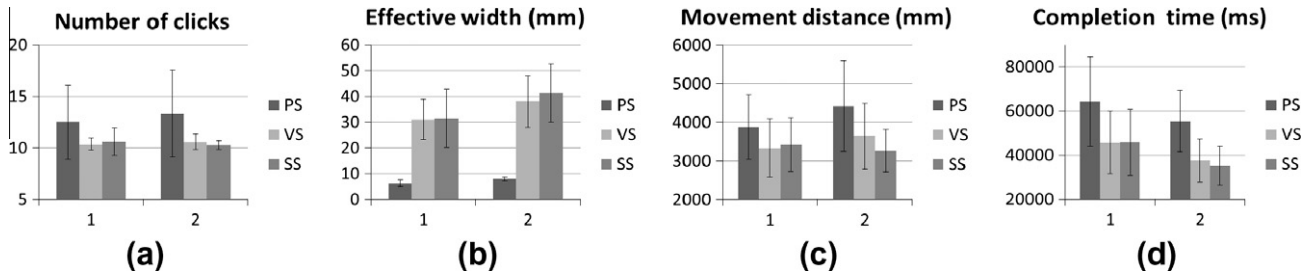

Fig. 14. Block charts for results about arrangements: (a) number of clicks; (b) effective width; (c) movement distance; and (d) completion time for arrangement 1 and arrangement 2.

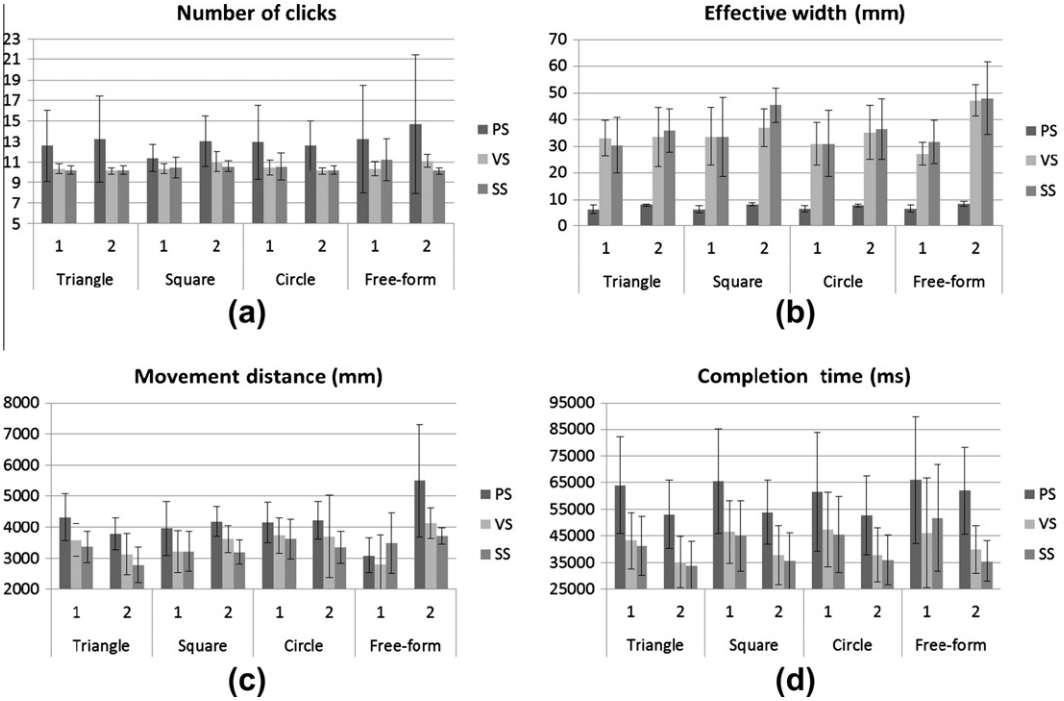

Fig. 15. Block charts for results about arrangements and path types: (a) number of clicks; (b) effective width; (c) movement distance; and (d) completion time for arrangement 1 and arrangement 2.

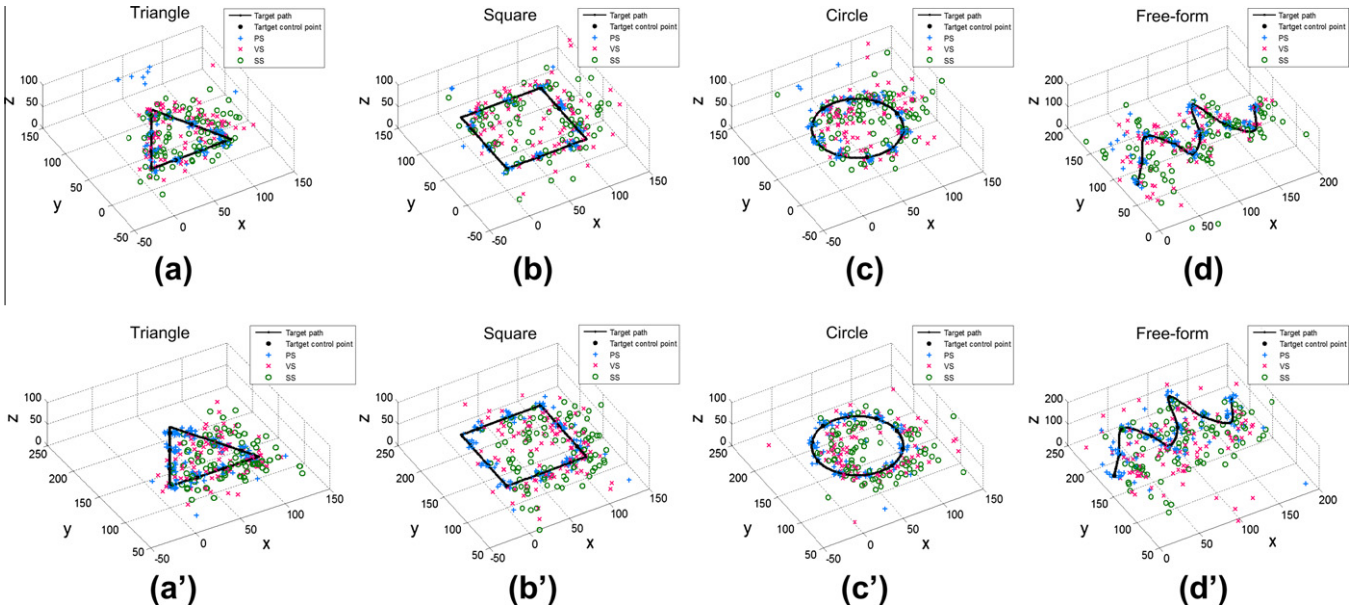

Fig. 16. Accumulated distribution charts of the initial selection position of the manipulation prop. (a–d) arrangement 1 and (a′–d′) arrangement 2.

longer distance than the SS method in arrangement 2 (post hoc analysis. $p = 0.086$). Considering the arrangements and path types, the SS showed significant reduced movement distance in the triangle path ($t_{(16)} = 2.283$, $p = 0.036$), but the PS ($t_{(16)} = -3.803$, $p = 0.002$) and VS ($t_{(16)} = -3.750$, $p = 0.002$) methods produced significantly increased movement distances in the free-form case of arrangement 2.

This could be because the SS methods frequently use the direction score rather than the other scores and the need to move the manipulation prop to select a control point in the free-form curve of the 1st arrangement.

4.3.2.4. Task completion time. The VS and SS methods recorded significantly shorter task completion times than the PS method. The SS method generally took less time than the VS method except with the first arrangement of the free-form curve (post hoc analysis. $p = 0.128$). On the other hand, the completion time of SS of free-form in arrangement 2 was significantly reduced compared to arrangement 1 ($t_{(10.337)} = 2.287$, $p = 0.044$). This could be because of significantly increased effective distances.

4.3.2.5. User subjective ratings. Table 6 show results of the user subjective ratings of experiment 2. The PS method showed the worst

Table 6
User subjective ratings of experiment 2 using Likert scales (1–7 score; 1 being most negative and 7 being most positive, # users assigning the rank. A total of 18 participants participated in our evaluation.).

Item		1	2	3	4	5	6	7	Average
Accuracy	PS	1	1	3	6	4	2	1	4.2
	VS	0	0	1	4	6	6	1	5.1
	SS	0	0	1	4	4	7	2	5.3
Speed	PS	1	4	5	4	4	0	0	3.3
	VS	0	0	0	1	8	8	1	5.5
	SS	0	0	1	0	8	6	3	5.6
Easiness	PS	2	2	6	4	2	2	0	3.4
	VS	0	0	0	2	7	9	0	5.4
	SS	0	0	2	1	7	4	4	5.4
Comfort	PS	1	2	6	5	2	1	1	3.7
	VS	0	0	2	2	7	6	1	5.1
	SS	0	0	1	6	3	5	3	5.2
Understandability	PS	1	0	1	4	4	4	4	5.1
	VS	0	0	0	3	4	7	4	5.7
	SS	0	0	1	3	2	5	7	5.8

ratings in most items with significant differences to the VS and SS conditions. The SS generally reported similar or a slightly more positive ratings than VS, but there were no significant differences between them.

4.3.2.6. Weighted values for each score. Fig. 17 shows an example and experimental result of the score based control point selection. The weighted value of α, β, γ, and ω were set to 0.25, as we mentioned in the implementation section. In the test condition, three control points (CP_0, CP_1, CP_2) were arranged, and each score value was recorded for 50 frames to examine how the proposed method worked (Fig. 17b–e). The still images with a 10 frame interval are shown in Fig. 17a.

In frame 1, the CP_0 was the closest to the manipulation prop with similar direction, high frequency score, and high time score. At that time, if a user gradually rotated the manipulation prop in a clockwise direction and moved it to the right, the CP_1 direction score increased, and distance scores for both CP_1 and CP_2 also relatively increased. At the same time, the frequency and time scores also increased. Afterward, CP_1 was selectable in the 16–30 frame range.

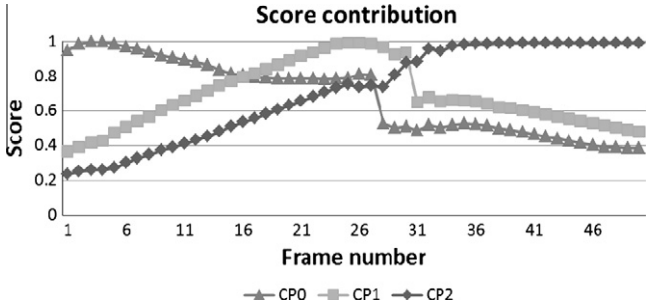

Fig. 18. Value of the score contribution for the test image sequences.

Fig. 18 shows the score contributions of the above test image sequences. CP_0 was selectable in the initial step, and CP_1 was selectable from frame 17 to frame 36. After that, CP_2 was selectable. The score-based selection method (SS) can increase the chance of selection of CP_1 compared with the Voronoi volume selection method (VS), which uses a distance measure.

4.3.2.7. Findings. The SS method was generally quicker than the VS method in most cases except the free-form path of arrangement 1. The SS in arrangement 2 had significant enhancements in the effective distance and completion time than in arrangement 1, and performed better than the VS method. Some of the comments from the participants were: "the SS was the most natural, especially with the farther away control points, but sometimes it was difficult to select a closer control," "The SS is easier to understand, it's more intuitive. However, when being used, the VS and SS seemed to be same," and "It was difficult to know which direction the points were relative to the manipulation prop." To enhance performance of the SS, we should visualize the effect of distance or direction score using semi-transplant volumes. We should also think about automatic weight values on various arrangements (Steed, 2006).

Considering the free-form path, the error was drastically higher than the other path shapes. To reduce error, participants were required to view it from multiple viewpoints but this was frequently

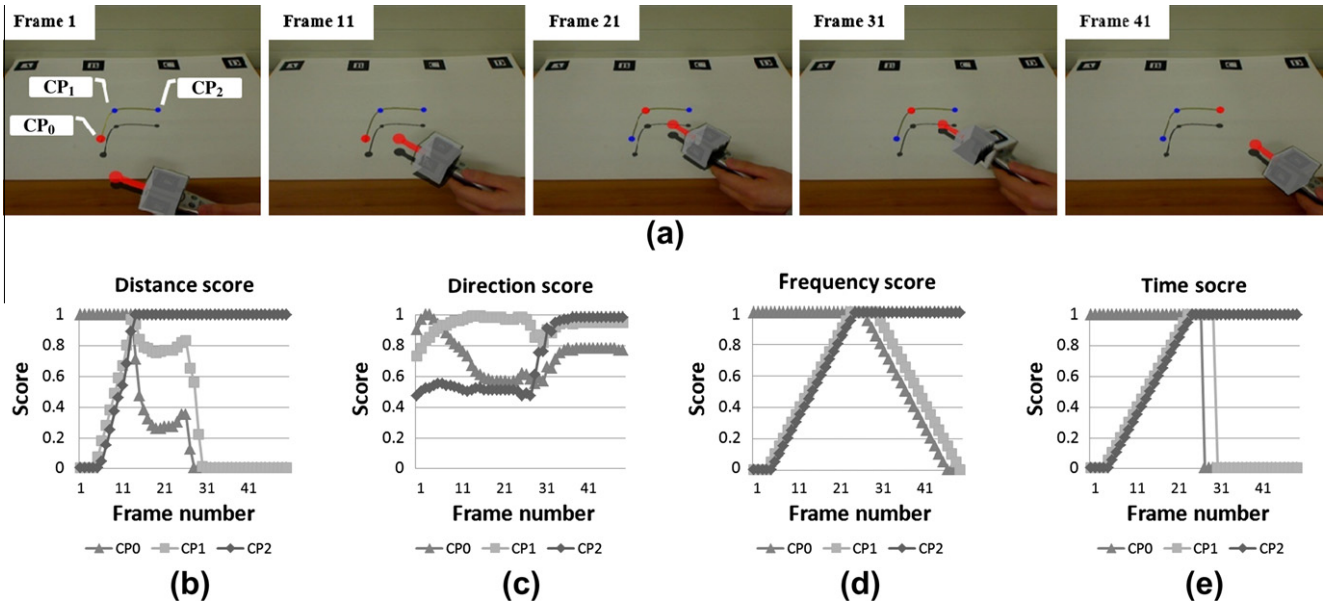

Fig. 17. An example and experimental result of the score-based control point selection method: (a) test image sequences for total 50 frames (still images at a 10-frame interval); (b) distance score; (c) direction score; (d) frequency score; (e) time score.

overlooked and it was hard to perceive depth when viewing the 3D models, even though shadow and occlusion cues were provided. We need a method to enhance the 3D depth perception, such as using a stereo HMD.

Sometimes, it was hard to turn the manipulation prop, so distance was the most important in both cases. This could be because the grip/angular lever principles would affect the performance (Grandjean, 1969; NASA, 2010): the ROM for the ulnar and radial bend of the wrist (i.e., the upward and downward movement of the wrist when the palm is facing sideways) is known to be between 15° and 30°, respectively. On the pronation and supination (i.e., the user rotates hands and forearms medially/laterally), each maximum angle is 65°/60°. On the flexion and extension (i.e., the user bends the wrist in a palmar/dorsal direction), each maximum angle is 60°/45°. To cover this limitation, we could consider a control gain that implicitly expands the user's wrist angle to cover large angles (e.g., wrist angle changes of 10° can cover selection direction angle of 20° on if control gain value is 2).

4.3.2.8. The weight values for the SS method. Again, the SS can have more flexible and scalable features to enhance performance because the SS can set a priority score with weighted values depending on various situations (e.g., the arrangement of the object, user's prop control ability, hardware performance, application). In Eq. (13), for the distance score, if a user has a good ability to use the manipulation prop accurately close to the target object's position, then it is better to increase the score value. On the other hand, if the weight value of the direction score is higher, then the method can be similar to that of the ray casting-based selection method, in which an object at a far distance can be selected. Especially considering the direction of the manipulation prop (mainly forward), it is better if target objects are further positioned from the user's hand. Considering the frequency score, if the user can quickly move a manipulation prop or a 3D object is dynamically moving, then the distance and direction score are frequently/continuously changed. In this case, the number of control points is more important than the distance/direction aspects; therefore, weighting the frequency score more could provide better performance. For similar reasons, the time score can be important in the dynamic running environment.

In Eq. (14), we have set both the decreasing ratio of the total sum to the previous score $C_{s,}$ and the increasing ratio of the total sum to the current score C_g, to a value of 0.5. These values can be adjusted to control the update speed in various situations. For example, in the case of a user's inability to control the prop or rapid head movement with a camera, sudden low-accuracy tracking, or moving 3D objects, by increasing the C_s, the previously selected control point is kept longer and the current selection can be more stable, but this can show a slightly late response. On the other hand, when the user has good control ability, stable tracking performance, or static object selection, a higher score weight C_g could be appropriate and contribute to a fast response effect.

5. Conclusion

In this paper, we present one of the first explorations of a 3D movement path manipulation in an AR environment. We use tangible AR-based enhanced movement path generation and manipulation methods that are different from the conventional GUI-based path authoring techniques that use keyboard or mouse input. In our case, AR authoring is conducted in an AR environment using a 3D UI to enhance spatial intuitiveness. The authoring and testing is done simultaneously and immediate modification is possible. Therefore a user can naturally (seamlessly) make a path in the AR environment.

We conducted two experiments, and from the first found that a path could be generated with Catmull–Rom spline-based method with significantly fewer control points than conventional methods, but with the same error and completion time. In the second path manipulation experiment, a movement path can be rapidly manipulated with relatively reduced manipulation prop movements using increased effective distance with the proposed dynamic control point selection method using four score measures. The method can also have more flexible and scalable features to enhance performance by setting a priority score with weighted values depending on various situations.

Through the suggested path generation and manipulation methods, a conventional static 3D object can be given movement properties. We believe that this can help create more interactive AR applications. The suggested methods are also applicable to AR environment-based path editing methods for sketching (Baudel, 1994), education (Ha et al., 2010), game, design (Matthew and Arvin, 2009), 3D modeling (Casper and Amnon, 1997; Kameyama, 1997; Nishino et al., 1998; Michalik et al., 2002), animation, and simulation applications.

In future work, we will conduct additional experiments with various movement paths, more spare or dense control points, and random selection sequences (selection sequences could make a significant difference between the VS and SS methods). We will also study a group selection method for simultaneously selecting multiple control points, because individual selection and manipulation of control points can be cumbersome for overall movement path manipulation. We also need a method for easily adding and removing control points while preserving details and nuances of the movement path. We will speed up the control point selection with the highest score among several candidates, because this method could cause a slight time delay in calculating the various score values. Finally, animation methods exploiting inverse kinematics, or signal processing (motion blending, warping, and displacement mapping) (Gleicher, 2001), will be added to create various interesting movements.

Acknowledgement

This work was supported by the Global Frontier R&D Program on Human-centered Interaction for Coexistence funded by the National Research Foundation of Korea grant funded by the Korean Government (MEST).

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.intcom.2011.06.006.

References

Bartels, R.H., Beatty, J.C., Barsky, B.A., 1987. An Introduction to Splines for Use in Computer Graphics and Geometric Modeling. Morgan Kaufmann Publishers, Inc.

Baudel, T., 1994. A mark-based interaction paradigm for free-hand drawing. ACM UIST 22, 185–192.

Billinghurst, M., Kato, H., Poupyrev, I., 2001. Collaboration with tangible augmented reality interfaces. HCII 22, 234–241.

Bowman, D., Hodges, L., 1997. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. ACM I3D, 35–38.

Bowman, D.A., Kruijff, E., LaViola, J.J., Poupyrev, I., 2004. 3D User Interfaces Theory and Practice. Addison-Wesley.

Buxton, W., Fitzmaurice, G., Balkrishnan, R., Kurtenbach, G., 2000. Large displays in automotive design. IEEE Computer Graphics and Applications, 68–75.

Casper, G.C., Amnon, A.C., 1997. Sketch input for conceptual surface design. Computers in Industry 34 (1), 125–137.

Catmull, E., Rom, R., 1974. A Class of Local Interpolating Splines. Computer Aided Geometric Design. Academic Press.

Clark, J.H., 1976. Designing surfaces in 3-D. ACM Communications 19 (8), 454–460.

Fitts, P.M., 1964. The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology 47, 381–391.
Forsberg, A., Herndon, K., Zeleznik, R., 1996. Aperture based selection for immersive virtual environments. ACM UIST, 95–96.
Geiger, C., Klompmaker, F., Stoecklein, J., Fritze, R., 2007. Development of an augmented reality game by extending a 3D authoring system. ACE, 230–231.
Gleicher, M., 2001. Motion path editing. In: ACM Symposium on Interactive 3D Graphics (I3D), pp. 195–202.
Grandjean, E., 1969. Fitting the Task to the Man: An Ergonomic Approach. Taylor and Francis.
Grossman, T., Balakrishnan, R., 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In: The SIGCHI Conference on Human Factors in Computing Systems (CHI), pp. 281–290.
Grossman, T., Balakrishnan, R., 2006. The design and evaluation of selection techniques for 3D volumetric displays. ACM UIST, 3–12.
Ha, T., Woo, W., 2010. An empirical evaluation of virtual hand techniques for 3D object manipulation in a tangible augmented reality environment. IEEE 3D User Interfaces, 91–98.
Ha, T., Lee, Y., Lee, J., Choi, H., Ryu, J., Lee, K., Woo, W., 2010. ARtalet: tangible user interface based immersive augmented reality authoring tool for Digilog book. ISUVR, 40–43.
Ha, T., Lee, Y., Woo, W., 2011. Digilog book for temple bell tolling experience based on interactive augmented reality with culture technology. Virtual Reality, Springer 15 (4), 295–309.
Haan, G.D., Koutek, M., Post, F.H., 2005. IntenSelect: using dynamic object rating for assisting 3D object selection. Virtual Environments, 201–209.
Han, L., Conti, G., 2005. A novel Approach to Free-form 3D Curve Recognition and Over-Sketching. Convegno Matematica, Arte e Industria Culturale, pp. 19–21.
Igarashi, T., Matsuoka, S., Tanaka, H., 1999, Teddy: a sketching interface for 3D freeform design. ACM SIGGRAPH, pp. 409–416.
Ishii, H., Ullmer, B., 1997. Tangible bits: towards seam-less interfaces between people, bits and atoms. ACM CHI, 234–241.
Kameyama, K., 1997. Virtual clay modeling system. ACM VRST, 197–200.
Kato, H., Billinghurst, M., 1999. Marker tracking and HMD Calibration for a video-based augmented reality conferencing system. In: The 2nd International Workshop on Augmented Reality (IWAR).
Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., Tachibana, K., 2000. Virtual object manipulation on a table-top AR environment. ISAR, 111–119.
Kato, H., Tachibana, K., Tanabe, M., Nakajima, T., Fukuda, Y., 2003. MagicCup: a tangible interface for virtual object manipulation in table-top augmented reality. In: The Augmented Reality Toolkit Workshop, pp. 75–76.
Krüger, W., Bohn, C.A., Fröhlich, B., Schüth, H., Strauss, W., Wesche, G., 1995. The responsive workbench: a virtual work environment. IEEE Computer 28 (7), 42–48.
Lee, G., Nelles, C., Billinghurst, M., Kim, G.J., 2004. Immersive authoring of tangible augmented reality applications. ISMAR, 172–181.
Liang, J., Green, M., 1994. JDCAD: a highly interactive 3D modeling system. Computers and Graphics 18 (4), 499–506.
Looser, J., Grasset, R., Seichter, H., Billinghurst, M., 2006. OSGART – a pragmatic approach to MR. In: Industrial Workshop, IEEE and ACM International Symposium on Mixed and Augmented Reality.
MacIntyre, B., Gandy, M., Dow, S., Bolter, J., 2004. DART: a toolkit for rapid design exploration of augmented reality experiences. ISMAR, 172–181.
Matthew, T.C., Arvin, A., 2009. A survey of sketch-based 3-D modeling techniques. Interacting with Computers 21 (3), 201–211.
Michalik, P., Kim, D., Bruderlin, B.D., 2002. Sketch- and constraint-based design of B-spline surfaces. ACM SMA, 297–304.
NASA, 2010. The Human Integration Design Handbook (HIDH), NASA/SP-2010-340. Section 4.4 RANGE OF MOTION and Appendix B.
Nishino, H., Utsumiya, K., Korida, K., 1998. 3D object modeling using spatial and pictographic gestures. VRST, 51–58.
Oda, O., Lister, L., White, S., Feiner, S., 2008. Developing an augmented reality racing game. INTETAIN, 8–10.
Olwal, A., Benko, H., Feiner, S., 2003. SenseShapes: using statistical geometry for object selection in a multimodal augmented reality system. In: IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR), pp. 300–301.
Pierce, J., Forsberg, A., Conway, M., Hong, S., Zeleznik, R., 1997. Image plane interaction techniques in 3D immersive environments. ACM I3D, 39–43.
Sachs, E., Roberts, A., Stoops, D., 1991. 3-Draw: a tool for designing 3D shapes. IEEE Computer Graphics and Applications, 18–26.
Schkolne, S., Pruett, M., Schröder, P., 2001. Surface drawing: creating organic 3D shapes with the hand and tangible tools. SIGCHI, 261–268.
Shaw, C., Green, M., 1997. THRED: a two-handed design system. Multimedia Systems Journal 5 (2), 126–139.
Shirley, P., Ashikhmin, M., Gleicher, M., Marschner, S., Reinhard, E., Sung, K., Thompson, W., Willemsen, P., 2005. Fundamentals of Computer Graphics. A.K. Peters, Ltd.
Steed, A., 2006. Towards a general model for selection in virtual environments. 3D User Interfaces (3DUI), 103–110.
Vanacken, L., Grossman, T., Coninx, K., 2009. Multimodal selection techniques for dense and occluded 3D virtual environments. International Journal of Human Computer Studies 67, 237–255.
Wesche, G., Droske, M., 2000. Conceptual free-form styling on the responsive workbench. VRST, 83–91.
Wesche, G., Seidel, H.P., 2001. FreeDrawer – a free-form sketching system on the responsive workbench. ACM VRST, 167–174.