4
Interaction techniques for direct 3D manipulation

Interaction techniques can be defined as [14] *the methods used to accomplish a given task via the given interface.*

Interaction techniques include both hardware and software components. Software components translate (map) input information captured by these input devices into system actions, which in their turn exercise an effect on the geometry of the 3D scene.

Captured input can include information such as the path followed by the hand moving through space, or a button pressed. This input is then transformed into a desired manipulation action, such as selecting, translating, rotating or scaling a virtual 3D object.

4.1
Selecting virtual 3D objects

Selecting a virtual 3D object is *the task of acquiring or identifying a particular object from the entire set of objects available* [14]. The parameters of the selection task include:

1. Vector **User** $\longrightarrow$ **Target 3D object** (i.e. distance and direction)
2. **Size** of the target 3D object
3. **Density of 3D objects** around the target 3D object
4. **Number of target 3D objects** to be selected
5. **Occlusion** of the target 3D object.

Further, each selection operation consists of the following three sub-operations (or subtasks), which decompose even further:

1. **User indicates the 3D object** (that is, user points at the 3D object):
   - by **occluding** the object
   - by **touching** the object — can be through a list, voice selection, automatic or iconic objects.

   - by **pointing** at the object — can be 2D, through 3D gaze, or 3D hand.
   - by **selecting** the object indirectly

2. **User confirms the selection** (that is, user triggers the selection):

   - by triggering some **event**, for example by pressing a button
   - by doing a **hand gesture**, for example by touching mutually two fingertips
   - by saying a **voice command**, for example "SELECT!"
   - by **no explicit command** — in this case it suffices to move the pointer into the object.

3. **User obtains feedback** from the system whether the 3D object has been selected or not through:

   - **text/symbolic** feedback, for example by printing out the message "OBJECT X SELECTED" to standard output (console)
   - **aural (audio)** feedback, by playing back a sound
   - **visual** feedback, for example by coloring the object in a distinct color
   - **force/tactile** feedback, for example by shaking the haptic hand.

4.2 Translating virtual 3D objects

Translating a virtual 3D object is *the task of changing the 3D position of an object* [14]. Before we can translate an object it has to be selected (see Section 4.1). The parameters of the translation task include:

1. Vector **User** $\longrightarrow$ **Initial position** (i.e. distance and direction)

2. Vector **User** $\longrightarrow$ **Target position** (i.e. distance and direction)

3. **Precision** required for translation.

Given these parameters, the translation vector **Initial position** $\longrightarrow$ **Target position** is then the difference between the second and the first vector above, constrained by the given precision **Precision**.

Since the target 3D object has been selected, now the user's hand motion can be mapped directly to the user's virtual hand motion in the VE, which therefore also translates the selected 3D object. Various mappings between the user's real hand and virtual hand are possible:

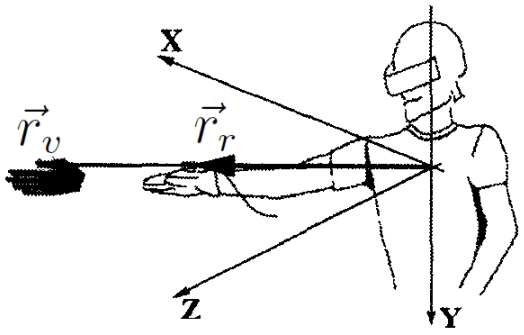

Figure 4.1: Go-go technique extends the hand non-linearly for $\vec{r} > \vec{r}_r$

– **Classical (simple) virtual hand**. Here the relation between the the virtual hand's position $\vec{p}_v = (x_v, y_v, z_v)$ and the real hand's position $\vec{p}_r = (x_r, y_r, z_r)$ is a simple one:

$$\vec{p}_v = \alpha \vec{p}_r$$

where α is a scaling factor between the real and virtual coordinate systems.

– **Go-Go technique** by Poupyrev *et al.* [15], see Figure 4.1. Instead of two Cartesian coordinate systems, here we use two 3D polar coordinate systems, one of the VE and the other one of the physical world, and both have the origin at the same point — in the centre of the user (for example, in the centre of the user's head or at the centre of the user's torso). Here a 3D point is determined by the coordinate tuple (r, ϕ, θ) which in turn defines the radius vector $\vec{r}$.

If the vector $\vec{r}_r = (r_r, \phi_r, \theta_r)$ points from the origin to the hand in the physical world, then the corresponding radius vector $\vec{r}_v = (r_v, \phi_v, \theta_v) = (r_v, \phi_r, \theta_r)$ (note that the angles stay the same) in the VE coordinate system is

$$r_v = \begin{cases} r_r & \text{if } r_r \leq D \\ r_r + \alpha(r_r - D)^2 & \text{otherwise} \end{cases}$$

where α is a scaling factor between the real and virtual coordinate systems and D is some threshold distance. The formula above expresses the fact that when the physical hand is close to the body, its movements correspond linearly to the virtual hand's movements. However if the user extends his hand beyond distance D, the arm's length grows at quadratic rate, thus enabling the user to translate the object to remote positions.

– **Other translation techniques**. Besides these two mappings between the user's real hand and virtual hand, which enable us to translate a 3D object, other translation techniques are available however are not relevant for this exposition. These techniques include World-in-Miniature [16]

where the user can manipulate iconic (shrank, downsized) representations of 3D objects, "Stretch Go-Go" [17] and ray-casting techniques where the translation is not hand-centered, but relative to the hand-object axis.

4.3 Rotating virtual 3D objects

Rotating a virtual 3D object is *the task of changing the orientation of an object* [14]. Before we can rotate an object it has to be selected (see Section 4.1). The parameters of the rotation task include:

1. **Distance** to target 3D object
2. **Initial rotation**
3. **Final rotation** or **Amount of rotation**
4. **Precision** required for rotation.

Given these parameters, we rotate the select object by either the angle **Final rotation** − **Initial rotation** or the angle **Amount of rotation**, starting at the angle **Initial rotation**, and respecting the given precision **Precision**.

Since the target 3D object has been selected, now the user's hand orientation can be mapped directly to the user's virtual hand orientation in the VE, which therefore also rotates the selected 3D object. For this, we use the same mapping as discussed in Section 4.2.

4.4 Scaling virtual 3D objects

Scaling a virtual 3D object is *the task of adjusting an object, according to a scale*. Before we can scale an object it has to be selected (see Section 4.1). Mine [18] lists two types of scaling:

- **Uniform scaling** — here the selected object is scaled by the same factor along all three extents (dimensions) equally.
- **Non-uniform scaling** — here the selected object is scaled along each dimension separately.

Mine [18] also lists two key parameters for a scaling operation:

- **Scaling factor** — this is the real number which multiplies one (in the case of non-uniform scaling), or all three dimensions (in the case of uniform scaling) of the object being scaled. The scaling factor can be:

  - Hand-controlled
  - Controlled through a physical control
  - Controlled through a virtual control

- **Center of scaling** — determines the behaviour of the scaling operation. It is the point which all objects move towards when you scale down and all points move away from when scaling up. Center of scaling can be on/in the:

  - **Object** — for object centered scaling, the center of scaling is defined to be the center of the selected object.
  - **Hand** — in hand centered scaling, all selected objects will scale about the current location of the hand.
  - **User-defined point** — alternately, objects can scale about some user defined center of scaling. User defined centers of scaling can be specified via direct interaction (e.g. the user can grab some icon representing the center of scaling and move it about) or by using some remote agent which the user moves (via remote control) to specify the desired center of scaling.