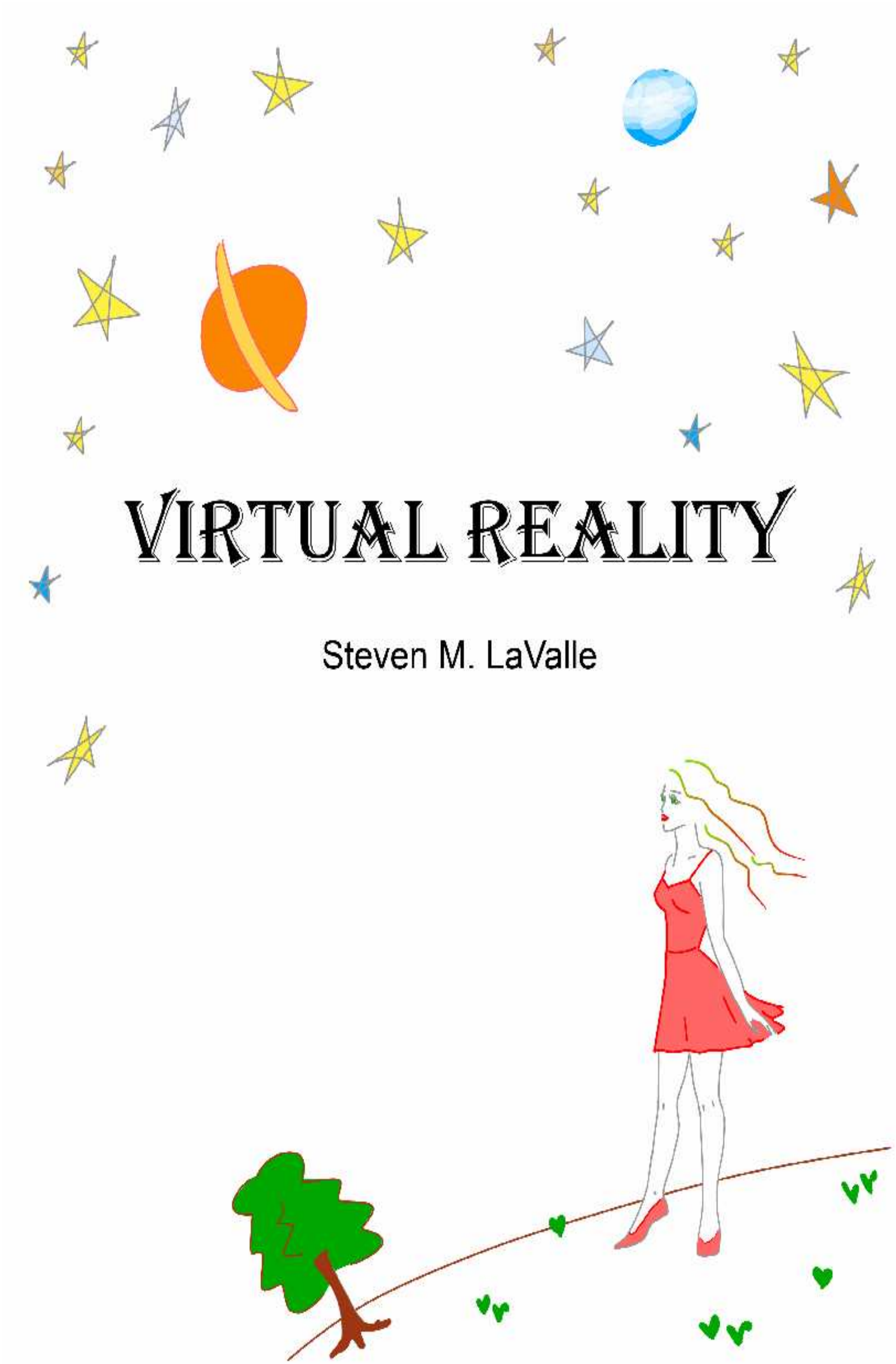


Chapter 10

Interaction

Steven M. LaValle

University of Oulu

Chapter 10

Interaction

How should users interact with the virtual world? How should they move about? How can they grab and place objects? How should they interact with representations of each other? How should they interact with files or the Internet? The following insight suggests many possible interfaces.

Universal Simulation Principle:
Any interaction mechanism from the real world can be simulated in VR.

For example, the user might open a door by turning a knob and pulling. As another example, the user operate a virtual aircraft by sitting in a mock-up cockpit (as was shown in Figure 1.16). One could even simulate putting on a VR headset, leading to an experience that is comparable to a dream within a dream!

In spite of the universal simulation principle, recall from Section 1.1 that the goal is not necessarily realism. It is often preferable to make the interaction *better than reality*. Therefore, this chapter introduces *interaction mechanisms* that may not have a counterpart in the physical world.

Section 10.1 introduces general motor learning and control concepts. The most important concept is *remapping*, in which a motion in the real world may be mapped into a substantially different motion in the virtual world. This enables many powerful interaction mechanisms. The task is to develop ones that are easy to learn, easy to use, effective for the task, and provide a comfortable user experience. Section 10.2 discusses how the user may move himself in the virtual world, while remaining fixed in the real world. Section 10.3 presents ways in which the user may interact with other objects in the virtual world. Section 10.4 discusses social interaction mechanisms, which allow users to interact directly with each other. Section 10.5 briefly considers some additional interaction mechanisms, such as editing text, designing 3D structures, and Web browsing.

10.1 Motor Programs and Remapping

Motor programs Throughout our lives, we develop fine motor skills to accomplish many specific tasks, such as writing text, tying shoelaces, throwing a ball, and riding a bicycle. These are often called *motor programs*, and are learned through repetitive trials, with gradual improvements in precision and ease as the amount of practice increases [22]. Eventually, we produce the motions without even having to pay attention to them. For example, most people can drive a car without paying attention to particular operations of the steering wheel, brakes, and accelerator.

In the same way, most of us have learned how to use interfaces to computers, such as keyboards, mice, and game controllers. Some devices are easier to learn than others. For example, a mouse does not take long, but typing quickly on a keyboard takes years to master. What makes one skill harder to learn than another? This is not always easy to predict, as illustrated by the *backwards brain bicycle*, which was designed by Destin Sandlin by reversing the steering operation so that turning the handlebars left turns the front wheel to the right [4]. It took Sandlin six months learn how to ride it, and at the end he was unable to ride an ordinary bicycle. Thus, he unlearned how to ride a normal bicycle at the expense of learning the new one.

Design considerations In the development of interaction mechanisms for VR, the main considerations are:

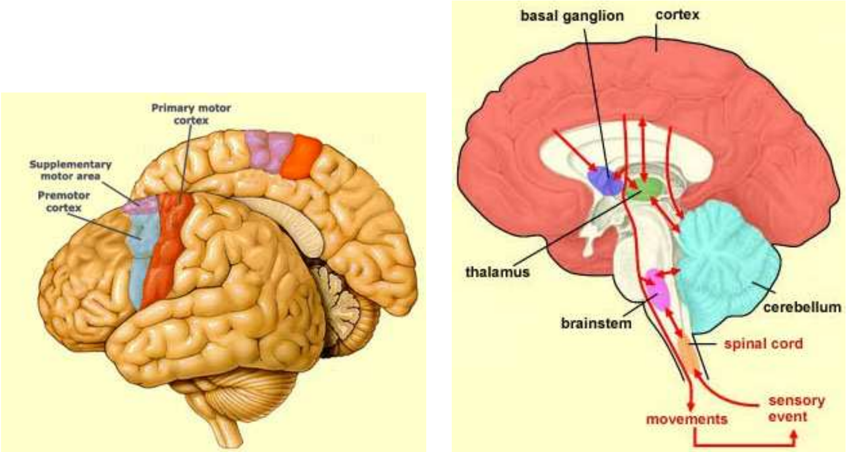
1. Effectiveness for the task in terms of achieving the required speed, accuracy, and motion range, if applicable.

2. Difficulty of learning the new motor programs; ideally, the user should not be expected to spend many months mastering a new mechanism.

3. Ease of use in terms of cognitive load; in other words, the interaction mechanism should require little or no focused attention after some practice.

4. Overall comfort during use over extended periods; the user should not develop muscle fatigue, unless the task is to get some physical exercise.

To design and evaluate new interaction mechanisms, it is helpful to start by understanding the physiology and psychology of acquiring the motor skills and programs. Chapters 5 and 6 covered these for visual perception, which is the process of converting sensory *input* into a perceptual experience. We now consider the corresponding parts for generating *output* in the form of body motions in the physical world. In this case, the brain sends motor signals to the muscles, causing them to move, while at the same time incorporating sensory feedback by utilizing the perceptual processes.



(a) (b)

Figure 10.1: (a) Part of the cerebral cortex is devoted to motion. (b) Many other parts interact with the cortex to produce and execute motions, including the thalamus, spinal cord, basal ganglion, brain stem, and cerebellum. (Figures from www.thebrain.mcgill.ca.)

The neurophysiology of movement First consider the neural hardware involved in learning, control, and execution of voluntary movements. As shown in Figure 10.1(a), some parts of the cerebral cortex are devoted to motion. The *primary motor cortex* is the main source of neural signals that control movement, whereas the *premotor cortex* and *supplementary motor area* appear to be involved in the preparation and planning of movement. Many more parts are involved in motion and communicate through neural signals, as shown in Figure 10.1(b). The most interesting part is the *cerebellum*, meaning "little brain", which is located at the back of the skull. It seems to be a special processing unit that is mostly devoted to motion, but is also involved in functions such as attention and language. Damage to the cerebellum has been widely seen to affect fine motor control and learning of new motor programs. It has been estimated to contain around 101 billion neurons [1], which is far more than the entire cerebral cortex, which contains around 20 billion. Even though the cerebellum is much smaller, a large number is achieved through smaller, densely packed cells. In addition to coordinating fine movements, it appears to be the storage center for motor programs.

One of the most relevant uses of the cerebellum for VR is in learning *sensorimotor relationships*, which become encoded into a motor program. All body motions involve some kind of sensory *feedback*. The most common example is *hand-eye coordination*; however, even if you move your arms with your eyes closed, proprioception provides information in the form of efference copies of the motor signals.

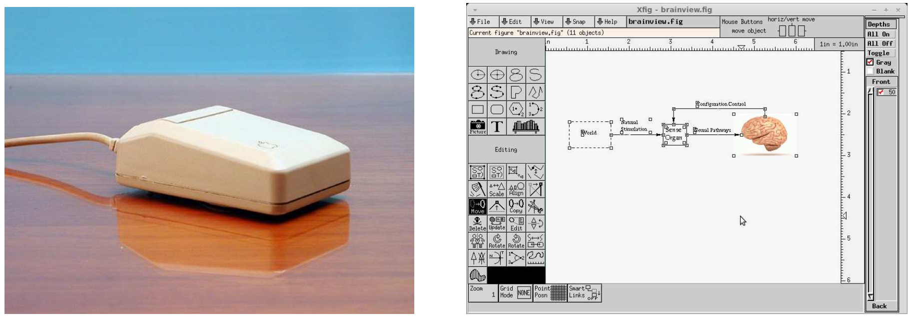
(a) (b)

Figure 10.2: (a) Atari 2600 Paddle controller. (b) The Atari Breakout game, in which the bottom line segment is a virtual paddle that allows the ball to bounce to the top and eliminate bricks upon contacts.

Developing a tight connection between motor control signals and sensory and perceptual signals is crucial to many tasks. This is also widely known in engineered systems, in which sensor-feedback and motor control are combined in applications such as robotics and aircraft stabilization; the subject that deals with this is called *control systems*. It is well-known that a *closed-loop* system is preferred in which sensor information provides feedback during execution, as opposed to *open-loop*, which specifies the motor signals as a function of time.

One of the most important factors is how long it takes to learn a motor program. As usual, there is great variation across humans. A key concept is *neuroplasticity*, which is the potential of the brain to reorganize its neural structures and form new pathways to adapt to new stimuli. Toddlers have a high level of neuroplasticity, which becomes greatly reduced over time through the process of *synaptic pruning*. This causes healthy adults to have about half as many synapses per neuron than a child of age two or three [17]. Unfortunately, the result is that adults have a harder time acquiring new skills such as learning a new language or learning how to use a complicated interface. In addition to the reduction of neuroplasticity with age, it also greatly varies among people of the same age.

Learning motor programs Now consider learning a motor program for a computer interface. A simple, classic example is the video game *Breakout*, which was developed by Atari in 1976. The player turns a knob, shown in Figure 10.2. This causes a line segment on the bottom of the screen to move horizontally. The Paddle contains a potentiometer that with calibration allows the knob orientation to be reliably estimated. The player sees the line segment positioned on the bottom of the screen and quickly associates the knob orientations. The learning process therefore involves taking information from visual perception and the proprioception signals from turning the knob and determining the sensorimotor



(a) (b)

Figure 10.3: (a) The Apple Macintosh mouse. (b) As a mouse moves across the table, the virtual finger on the screen moves correspondingly, but is rotated by 90 degrees and travels over longer distances.

relationships. Skilled players could quickly turn the knob so that they could move the line segment much more quickly than one could move a small tray back and forth in the real world. Thus, we already have an example where the virtual world version allows better performance than in reality.

In the Breakout example, a one-dimensional *mapping* was learned between the knob orientation and the line segment position. Many alternative control schemes could be developed; however, they are likely to be more frustrating. If you find an emulator to try Breakout, it will most likely involve using keys on a keyboard to move the segment. In this case, the amount of time that a key is held down corresponds to the segment displacement. The segment velocity is set by the program, rather than the user. A reasonable alternative using modern hardware might be to move a finger back and forth over a touch screen while the segment appears directly above it. The finger would not be constrained enough due to extra DOFs and the rapid back and forth motions of the finger may lead to unnecessary fatigue, especially if the screen is large. Furthermore, there are conflicting goals in positioning the screen: Making it as visible as possible versus making it comfortable for rapid hand movement over a long period of time. In the case of the Paddle, the motion is accomplished by the fingers, which have high dexterity, while the forearm moves much less. The mapping provides an association between body movement and virtual object placement that achieves high accuracy, fast placement, and long-term comfort.

Figure 10.3 shows a more familiar example, which is the computer mouse. As the mouse is pushed around on a table, encoders determine the position, which is converted into a pointer position on the screen. The sensorimotor mapping seems a bit more complex than in the Breakout example. Young children seem to immediately learn how to use the mouse, whereas older adults require some practice.

The 2D position of the mouse is mapped to a 2D position on the screen, with two fundamental distortions: 1) The screen is rotated 90 degrees in comparison to the table (horizontal to vertical motion. 2) The motion is scaled so that small physical motions produce larger screen motions. The advantages of the original Xerox Alto mouse were scientifically argued in [8] in terms of human skill learning and *Fitts's law* [15, 19], which mathematically relates pointing task difficulty to the time required to reach targets.

For a final example, suppose that by pressing a key, the letter “h” is instantly placed on the screen in a familiar font. Our visual perception system recognizes the “h” as being equivalent to the version on paper. Thus, typing the key results in the perception of “h”. This is quite a comfortable, fast, and powerful operation. The amount of learning required seems justified by the value of the output.

Motor programs for VR The examples given so far already seem closely related to VR. A perceptual experience is controlled by body movement that is sensed through a hardware device. Using the universal simulation principle, any of these and more could be brought into a VR system. The physical interaction part might be identical (you could really be holding an Atari Paddle), or it could be simulated through another controller. Think about possible designs.

Using the tracking methods of Chapter 9, the position and orientation of body parts could be reliably estimated and brought into VR. For the case of head tracking, it is essential to accurately maintain the viewpoint with high accuracy and zero effective latency; otherwise, the VR experience is significantly degraded. This is essential because the perception of stationarity must be maintained for believability and comfort. The motion of the sense organ must be matched by a tracking system.

Remapping For the motions of other body parts, this perfect matching is not critical. Our neural systems can instead learn associations that are preferable in terms of comfort, in the same way as the Atari Paddle, mouse, and keyboard work in the real world. Thus, we want to do *remapping*, which involves learning a sensorimotor mapping that produces different results in a virtual world than one would expect from the real world. The keyboard example above is one of the most common examples of remapping. The process of pushing a pencil across paper to produce a letter has been replaced by pressing a key. The term remapping is even used with keyboards to mean the assignment of one or more keys to another key.

Remapping is natural for VR. For example, rather than reaching out to grab a virtual door knob, one could press a button to open the door. For a simpler case, consider holding a controller for which the pose is tracked through space, as allowed by the HTC Vive system. A scaling parameter could be set so that one centimeter of hand displacement in the real world corresponds to two centimeters of displacement in the virtual world. This is similar to the scaling parameter for the mouse. Section 10.2 covers the remapping from natural walking in the real

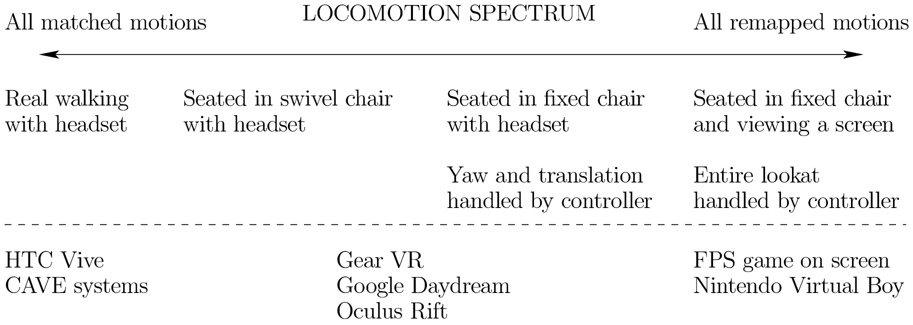


Figure 10.4: Moving from left to right, the amount of viewpoint mismatch between real and virtual motions increases.

world to achieving the equivalent in a virtual world by using a controller. Section 10.3 covers object interaction methods, which are again achieved by remappings. You can expect to see many new remapping methods for VR in the coming years.

10.2 Locomotion

Suppose that the virtual world covers a much larger area than the part of the real world that is tracked. In other words, the matched zone is small relative to the virtual world. In this case, some form of interaction mechanism is needed to move the user in the virtual world while she remains fixed within the tracked area in the real world. An interaction mechanism that moves the user in this way is called *locomotion*. It is as if the user is riding in a virtual vehicle that is steered through the virtual world.

Figure 10.4 shows a spectrum of common locomotion scenarios. At the left, the user walks around in an open space while wearing a headset. No locomotion is needed unless the virtual world is larger than the open space. This case involves no mismatch between real and virtual motions.

The two center cases correspond to a seated user wearing a headset. In these cases, an interaction mechanism is used to change the position of the matched zone in the virtual world. If the user is seated in a swivel chair, then he could change the direction he is facing (yaw orientation) by rotating the chair. This can be considered as orienting the user's torso in the virtual world. If the user is seated in a fixed chair, then the virtual torso orientation is typically changed using a controller, which results in more mismatch. The limiting case is on the right of Figure 10.4, in which there is not even head tracking. If the user is facing a screen, as in the case of a first-person shooter game on a screen, then a game controller is used to change the position and orientation of the user in the virtual world. This is the largest amount of mismatch because all changes in viewpoint are generated by the controller.

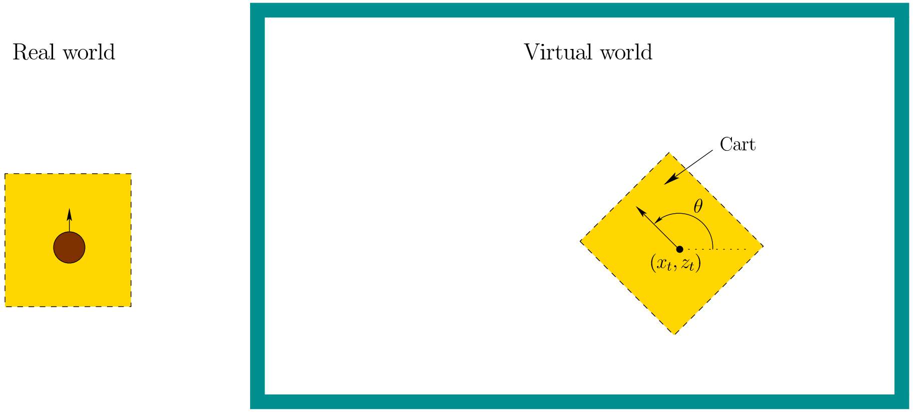


Figure 10.5: Locomotion along a horizontal terrain can be modeled as steering a cart through the virtual world. A top-down view is shown. The yellow region is the matched zone (recall Figure 2.15), in which the user's viewpoint is tracked. The values of x_t, z_t, and θ are changed by using a controller.

Redirected walking If the user is tracked through a very large space, such as a square region of at least 30 meters on each side, then it is possible to make her think she is walking in straight lines for kilometers while she is in fact walking in circles. This technique is called *redirected walking* [27]. Walking along a straight line over long distances without visual cues is virtually impossible for humans (and robots!) because in the real world it is impossible to achieve perfect symmetry. One direction will tend to dominate through an imbalance in motor strength and sensory signals, causing people to travel in circles.

Imagine a VR experience in which a virtual city contains long, straight streets. As the user walks down the street, the yaw direction of the viewpoint can be gradually varied. This represents a small amount of mismatch between the real and virtual worlds, and it causes the user to walk along circular arcs. The main trouble with this technique is that the user has free will and might decide to walk to the edge of the matched zone in the real world, even if he cannot directly perceive it. In this case, an unfortunate, disruptive warning might appear, suggesting that he must rotate to reset the yaw orientation.

Locomotion implementation Now consider the middle cases from Figure 10.4 of sitting down and wearing a headset. Locomotion can then be simply achieved by moving the viewpoint with a controller. It is helpful to think of the matched zone as a controllable cart that moves across the ground of the virtual environment; see Figure 10.5. First consider the simple case in which the ground is a horizontal plane. Let T_{track} denote the homogeneous transform that represents the tracked position and orientation of the cyclopean (center) eye in the physical world. The methods described in Section 9.3 could be used to provide T_{track} for the current time.

The position and orientation of the cart is determined by a controller. The homogeneous matrix:

$$T_{cart} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & x_t \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & z_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10.1}$$

encodes the position (x_t, z_t) and orientation θ of the cart (as a yaw rotation, borrowed from (3.18)). The height is set at $y_t = 0$ in (10.1) so that it does not change the height determined by tracking or other systems (recall from Section 9.2 that the height might be set artificially if the user is sitting in the real world, but standing in the virtual world).

The eye transform is obtained by chaining T_{track} and T_{cart} to obtain

$$T_{eye} = (T_{track}T_{cart})^{-1} = T_{cart}^{-1}T_{track}^{-1} \tag{10.2}$$

Recall from Section 3.4 that the eye transform is the *inverse* of the transform that places the geometric models. Therefore, (10.2) corresponds to changing the perspective due to the cart, followed by the perspective of the tracked head on the cart.

To move the viewpoint for a fixed direction θ, the x_t and z_t components are obtained by integrating a differential equation:

$$\begin{aligned} \dot{x}_t &= s\cos\theta \\ \dot{z}_t &= s\sin\theta. \end{aligned} \tag{10.3}$$

Integrating (10.3) over a time step Δt, the position update appears as

$$\begin{aligned} x_t[k+1] &= x_t[k] + \dot{x}_t\Delta t \\ z_t[k+1] &= z_t[k] + \dot{z}_t\Delta t. \end{aligned} \tag{10.4}$$

The variable s in (10.3) is the forward speed. The average human walking speed is about 1.4 meters per second. The virtual cart can be moved forward by pressing a button or key that sets $s = 1.4$. Another button can be used to assign $s = -1.4$, which would result in backward motion. If no key or button is held down, then $s = 0$, which causes the cart to remain stopped. An alternative control scheme is to use the two buttons to increase or decrease the speed, until some maximum limit is reached. In this case, motion is sustained without holding down a key.

Keys could also be used to provide lateral motion, in addition to forward/backward motion. This is called *strafing* in video games. It should be avoided, if possible, because it cases unnecessary lateral vection.

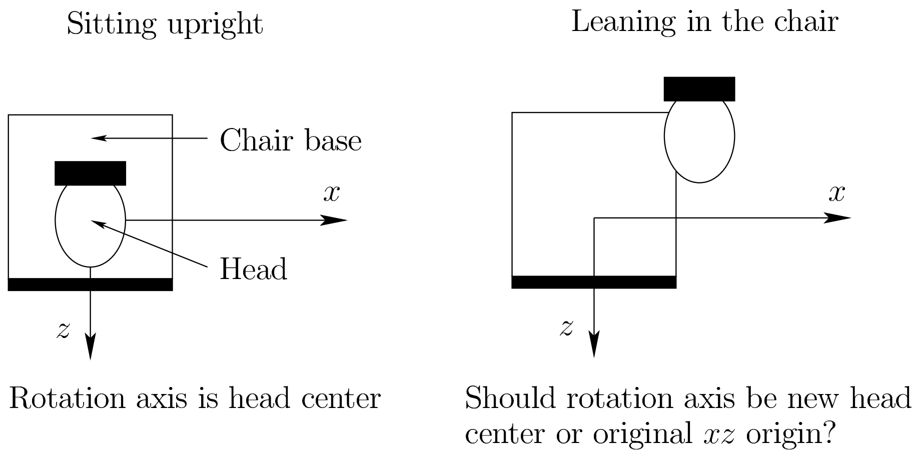


Figure 10.6: On the right the yaw rotation axis is centered on the head, for a user who is upright in the chair. On the left, the user is leaning over in the chair. Should the rotation axis remain fixed, or move with the user?

Issues with changing direction Now consider the orientation θ. To move in a different direction, θ needs to be reassigned. The assignment could be made based on the user's head yaw direction. This becomes convenient and comfortable when the user is sitting in a swivel chair and looking forward. By rotating the swivel chair, the direction can be set. (However, this could become a problem for a wired headset because the cable could wrap around the user.)

In a fixed chair, it may become frustrating to control θ because the comfortable head yaw range is limited to only 60 degrees in each direction (recall Figure 5.21). In this case, buttons can be used to change θ by small increments in clockwise or counterclockwise directions. Unfortunately, changing θ according to constant angular velocity causes yaw vection, which is nauseating to many people. Some users prefer to tap a button to instantly yaw about 10 degrees each time. If the increments are too small, then vection appears again, and if the increments are too large, then users become confused about their orientation.

Another issue is where to locate the center of rotation, as shown in Figure 10.6. What happens when the user moves his head away from the center of the chair in the real world? Should the center of rotation be about the original head center or the new head center? If it is chosen as the original center, then the user will perceive a large translation as θ is changed. However, this would also happen in the real world if the user were leaning over while riding in a cart. If it is chosen as the new head center, then the amount of translation is less, but might not correspond as closely to reality.

For another variation, the car-like motion model (8.30) from Section 8.3.2 could be used so that the viewpoint cannot be rotated without translating. In other words, the avatar would have a minimum turning radius. In general, the viewpoint could be changed by controlling any virtual vehicle model. Figure 1.1 from Chapter 1 showed an example in which the "vehicle" is a bird.

Vection reduction strategies The main problem with locomotion is vection, which leads to VR sickness. Recall from Section 8.4 that six different kinds of vection occur, one for each DOF. Furthermore, numerous factors were given that affect the sensitivity to vection. Reducing the intensity of these factors should reduce vection and, hopefully, VR sickness.

Several strategies for reducing vection-based VR sickness are:

1. If the field of view for the optical flow is reduced, then the vection is weakened. A common example is to make a cockpit or car interior that blocks most of the optical flow.

2. If the viewpoint is too close to the ground, then the magnitudes of velocity and acceleration vectors of moving features are higher. This is why you might feel as if you are traveling faster in a small car that is low to the ground in comparison to riding at the same speed in a truck or minivan.

3. Surprisingly, a larger mismatch for a short period of time may be preferable to a smaller mismatch over a long period of time; see Figure 10.7.

4. Having high spatial frequency will yield more features for the human vision system to track. Therefore, if the passing environment is smoother, with less detail, then vection should be reduced. Consider the case of traveling up a staircase. If the steps are clearly visible so that they appear as moving horizontal stripes, then the user may quickly come nauseated by the strong vertical vection signal.

5. Reducing contrast, such as making the world seem hazy or foggy while accelerating, may help.

6. Providing other sensory cues such as blowing wind or moving audio sources might provide stronger evidence of motion. Including vestibular stimulation in the form of a rumble or vibration may also help lower the confidence of the vestibular signal. Even using head tilts to induce changes in virtual-world motion may help because it would cause distracting vestibular signals.

7. If the *world* is supposed to be moving, rather than the user, then making it clear through cues or special instructions can help.

8. Providing specific tasks, such as firing a laser at flying insects, may provide enough distraction from the vestibular conflict. If the user is instead focused entirely on the motion, then she might become sick more quickly.

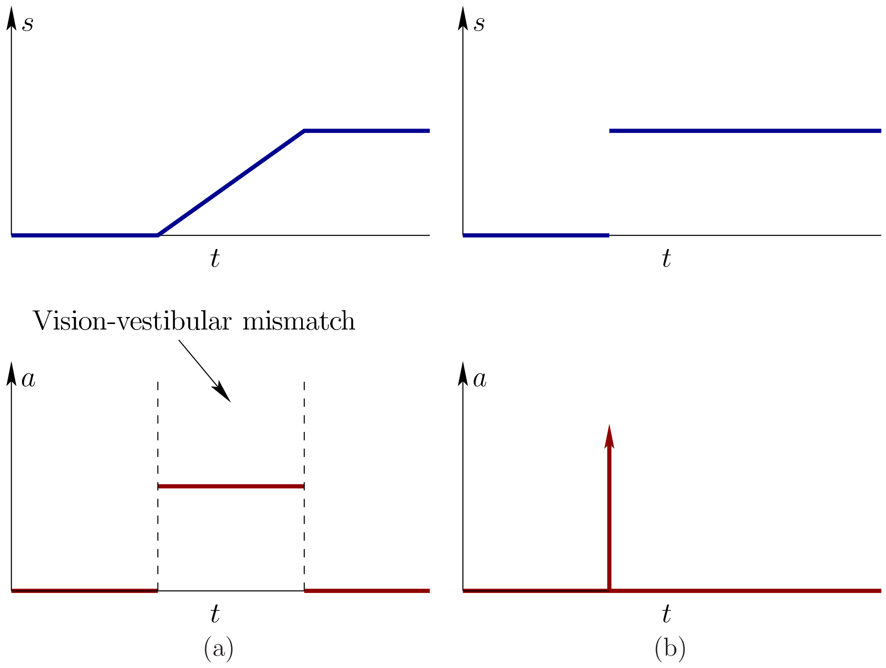


Figure 10.7: (a) Applying constant acceleration over a time interval to bring the stopped avatar up to a speed limit. The upper plot shows the speed over time. The lower plot shows the acceleration. The interval of time over which there is nonzero acceleration corresponds to a mismatch with the vestibular sense. (b) In this case, an acceleration impulse is applied, resulting in the desired speed limit being immediately achieved. In this case, the mismatch occurs over a time interval that is effectively zero length. In practice, the perceived speed changes in a single pair of consecutive frames. Surprisingly, most people consider case (b) to be more comfortable than (a). Perhaps the brain prefers an outlier mismatch for a very short time interval, as supposed to a smaller, sustained mismatch over a longer time interval (such as 5 seconds).

9. The adverse effects of vection may decrease through repeated practice. People who regularly play FPS games in front of a large screen already seem to have reduced sensitivity to vection in VR. Requiring users to practice before sickness is reduced might not be a wise strategy for companies hoping to introduce new products. Imagine trying some new food that makes you nauseated after the first 20 times of eating it, but then gradually becomes more acceptable. Who would keep trying it?

A final suggestion is to avoid locomotion wherever possible! Try to design experiences that do not critically depend on it.

Non-planar locomotion Now consider more complicated locomotion cases. If the user is walking over a terrain, then the y component can be simply increased or decreased to reflect the change in altitude. This may seem realistic, but keep in mind that it increases the amount of mismatch between the real and virtual worlds because vertical vection is combined with forward vection.

In the case of moving through a 3D medium, all six forms of vection from Section 8.4 become enabled. Common settings include a virtual spacecraft, aircraft, or scuba diver. Yaw, pitch, and roll vection can be easily generated. For example, imagine flying a virtual spacecraft. By rolling the craft, roll vection can be caused as the stars spin around in a circular pattern. If a developer must make a craft move in these ways, then the prior suggestions for reducing vection intensity should be followed. Furthermore, careful experimentation with human subjects should be performed to determine which forms of vection are worse in the particular application; see Chapter 12. To avoid singularities, for systems in which all 3 DOFs of rotational motion are possible, the virtual vehicle transformations are best maintained in terms of quaternions (recall the issues from Section 3.3).

Adding special effects that move the viewpoint will cause further difficulty with vection. For example, making an avatar jump up and down will cause vertical vection. It is also a bad idea to account for swaying head motions while walking because of the increased mismatch. Imagine a far worse case of looking out through the eyes of an avatar that performs gymnastics. The view of the world may become unbearable during multiple flips.

Specialized hardware Many kinds of hardware have been developed to support locomotion. One of the oldest examples is to create an entire cockpit for aircraft flight simulation (recall Figure 1.16). Figure 10.8(a) shows an *omnidirectional treadmill* that enables walking in any direction and over any distance. Exercise machines, such as a stationary bicycle have been connected to VR systems so that the user can pedal and steer to guide himself through a large virtual world, as shown in Figure 10.8(b). Figure 1.1 showed a mechanical platform for virtual flying like a bird.

(a) (b)

Figure 10.8: (a) An omnidirectional treadmill used in a CAVE system by the US Army for training. (b) A home-brew bicycle riding system connected to a VR headset, developed by Paul Dyan.

Teleportation The locomotion methods covered so far have mainly focused on reproducing experiences that are familiar in the real world, which provide instances of the universal simulation principle. In VR, however, we could also move in ways that are physical implausible. The most common is *teleportation*, which it works like a transporter in the TV series Star Trek. The user is immediately transported to another location.

How is the desired location determined? One simple mechanism is a *virtual laser pointer* (or *3D mouse*), which is accomplished by the user holding a controller that is similar in shape to a laser pointer in the real world. A smart phone could even be used. The user rotates the controller to move a laser dot in the virtual world. This requires performing a ray casting operation (recall from Section 7.1) to find the nearest visible triangle, along the ray that corresponds to the laser light.

To select a location where the user would prefer to stand, she could simply point the virtual laser and press a key to be instantly teleported. To make pointing at the floor easier, the beam could actually be a parabolic arc that follows gravity, similar to a stream of water; see Figure 10.9. Places that are not visible can be selected by using a pop-up map, or even performing a text-based search (voice commands could be used instead of typing). One method, called *world in miniature*, involves showing the user a virtual small-scale version of the environment [30]. This is effectively a 3D map.

Wayfinding The cognitive problem of learning a spatial representation and using it to navigate is called *wayfinding*. This is a higher-level process than the low-level locomotion mechanism, but the two are closely related. One trouble with locomotion systems that are not familiar in the real world is that users might not learn the spatial arrangement of the world around them. Would your

Figure 10.9: A virtual "laser pointer" that follows a parabolic arc so that a destination for teleportation can be easily specified as a point on the floor. (Image from the Budget Cuts game on the HTC Vive platform.)

brain still form place cells for an environment in the real world if you were able to teleport from place to place? We widely observe this phenomenon with people who learn to navigate a city using only GPS or taxi services, rather than doing their own wayfinding.

The teleportation mechanism reduces vection, and therefore VR sickness; however, it may come at the cost of reduced learning of the spatial arrangement of the environment. When performing teleportation, it is important not to change the yaw orientation of the viewpoint; otherwise, the user may become eve more disoriented. He might not understand where he is now positioned and oriented in the virtual world relative to the previous location.

Note that the universal simulation principle can once again be employed to borrow any effective navigation tools from the real world. If virtual buildings and cities are laid out in ways that are common in the real world, then they should be easier to navigate. Signs and landmarks can even be placed into the virtual world to help with navigation. In the real world, signs often tell us the locations of exits, the names of streets, or the boundary of a district. Landmarks such as tall buildings, windmills, or towers provide visual cues that are effective for navigation over long distances. Many of these ideas are discussed in Chapter 7 of [6].

Figure 10.10: Tom Cruise moving windows around on a holographic display in the 2002 movie Minority Report. It is a great-looking interaction mechanism for Hollywood, but it is terribly tiring in reality. The user would quickly experience *gorilla arms*.

10.3 Manipulation

We interact with objects in the real world for many reasons. You might eat a bowl of soup by moving a spoon between the bowl and your mouth. You might pick up a rock and throw it as far as possible. You might put on a pair of pants. These examples and many more fall under the topic of *manipulation*. In the real world, manipulation involves complex sensorimotor relationships which, through evolution and experience, enable us to manipulate objects under a wide variety of settings. The variation of objects includes differences in size, weight, friction, flexibility, temperature, fragility, and so on. Somehow our bodies can handle that. Getting robots to perform the manipulation in the ways that humans do has been a long and frustrating road, with only limited success [23].

Because of manipulation complexity in the real world, it is an ideal candidate for applying the remapping concepts from Section 10.1 to make manipulation as simple as possible in VR. The virtual world does not have to follow the complicated physics of manipulation. It is instead preferable to make operations such as selecting, grasping, manipulating, carrying, and placing an object as fast and easy as possible. Furthermore, extensive reaching or other forms of muscle strain should be avoided, unless the VR experience is designed to provide exercise.

Avoid gorilla arms One of the most common misconceptions among the public is that the interface used by Tom Cruise in the movie Minority Report is desirable; see Figure 10.10. In fact, it quickly leads to the well-known problem of *gorilla* *arms*, in which the user quickly feels fatigue from extended arms. How long can you hold your arms directly in front of yourself without becoming fatigued?

Selection One of the simplest ways to select an object in the virtual world is with the virtual laser pointer, which was described in Section 10.2. Several variations may help to improve the selection process. For example, the user might instead hold a virtual flashlight that illuminates potential selections. The field of view of the flashlight could be adjustable [16]. A virtual mirror could be placed so that a selection could be made around a corner. Chapter 5 of [6] offers many other suggestions.

With a pointer, the user simply illuminates the object of interest and presses a button. If the goal is to retrieve the object, then it can be immediately placed in the user's virtual hand or inventory. If the goal is to manipulate the object in a standard, repetitive way, then pressing the button could cause a virtual motor program to be executed. This could be used, for example, to turn a doorknob, thereby opening a door. In uses such as this, developers might want to set a limit on the depth of the laser pointer, so that the user must be standing close enough to enable the interaction. It might seem inappropriate, for example, to turn doorknobs from across the room!

If the object is hard to see, then the selection process may be complicated. It might be behind the user's head, which might require uncomfortable turning. The object could be so small or far away that it occupies only a few pixels on the screen, making it difficult to precisely select it. The problem gets significantly worse if there is substantial clutter around the object of interest, particularly if other selectable objects are nearby. Finally, the object may be partially or totally occluded from view.

Manipulation If the user carries an object over a long distance, then it is not necessary for her to squeeze or clutch the controller; this would yield unnecessary fatigue. In some cases, the user might be expected to carefully inspect the object while having it in possession. For example, he might want to move it around in his hand to determine its 3D structure. The object orientation could be set to follow exactly the 3D orientation of a controller that the user holds. The user could even hold a real object in hand that is tracked by external cameras, but has a different appearance in the virtual world. This enables familiar force feedback to the user, a concept that is revisited in Section 13.1. Note that an object could even be manipulated directly in its original place in the virtual world, without bringing it close to the user's virtual body [5]. In this case, the virtual hand is brought to the object, while the physical hand remains in place. Having a longer arm than normal can also be simulated [26], to retrieve and place objects over greater distances.

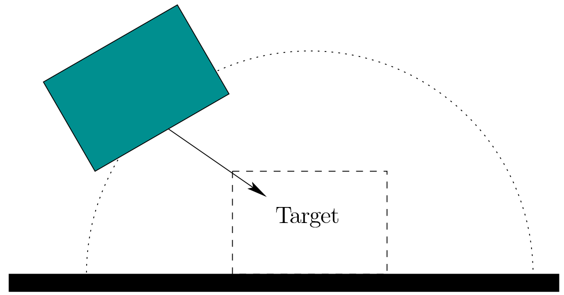


Figure 10.11: To make life easier on the user, a *basin of attraction* can be defined around an object so that when the basin in entered, the dropped object is attracted directly to the target pose.

Placement Now consider ungrasping the object and placing it into the world. An easy case for the user is to press a button and have the object simply fall into the right place. This is accomplished by a *basin of attraction* which is an attractive potential function defined in a neighborhood of the target pose (position and orientation); see Figure 10.11. The minimum of the potential function is at the target. After the object is released, the object falls into the target pose by moving so that the potential is reduced to its minimum. This behavior is seen in many 2D drawing programs so that the endpoints of line segments conveniently meet. An example of convenient object placement is in the 2011 Minecraft sandbox game by Markus Persson (Notch), in which building blocks simply fall into place. Children have built millions of virtual worlds in this way.

Alternatively, the user may be required to delicately place the object. Perhaps the application involves stacking and balancing objects as high as possible. In this case, the precision requirements would be very high, placing a burden on both the controller tracking system and the user.

Remapping Now consider the power of remapping, as described in Section 10.1. The simplest case is the use of the button to select, grasp, and place objects. Instead of a button, continuous motions could be generated by the user and tracked by systems. Examples include turning a knob, moving a slider bar, moving a finger over a touch screen, and moving a free-floating body through space. Recall that one of the most important aspects of remapping is easy learnability. Reducing the number of degrees of freedom that are remapped will generally ease the learning process. To avoid gorilla arms and related problems, a scaling factor could be imposed on the tracked device so that a small amount of position change in the controller corresponds to a large motion in the virtual world. This problem could again be studied using Fitts's law as in the case of the computer mouse. Note that this might have an adverse effect on precision in the virtual world. In some settings orientation scaling might also be desirable. In this case, the 3D angular

(a) (b)

Figure 10.12: (a) A pair of hand-held controllers that came with the HTC Vive headset in 2016; the device includes side buttons, a trigger, and a touch pad for the thumb. (b) A user trying the controllers (prototype version).

velocity $(\omega_x, \omega_y, \omega_z)$ could be scaled by a factor to induce more rotation in the virtual world than in the real world.

Current systems The development of interaction mechanisms for manipulation remains one of the greatest challenges for VR. Current generation consumer VR headsets either leverage existing game controllers, as in the bundling of the XBox 360 controller with the Oculus Rift in 2016, or introduce systems that assume large hand motions are the norm, as in the HTC Vive headset controller, as shown in Figure 10.12. Controllers that have users moving their hands through space seem not too far from the Minority Report interaction mechanism shown in Figure 10.10. Others are developing gesturing systems that involve no hardware in the hands, as in the Leap Motion system that was shown in Figure 9.25 from Section 9.4. These are perhaps updated versions of the vision of "goggles and gloves" that was popular in the 1990s (recall Figure 1.30(c) from Section 1.3). Rapid evolution of methods and technologies for manipulation can be expected in the coming years, with increasing emphasis on user comfort and ease of use.

10.4 Social Interaction

Communication and social interaction are vast subjects that extend well outside of the scope of this book. Furthermore, social interaction in VR, or *social VR*, remains in a stage of infancy, with substantial experimentation and rethinking of paradigms occurring. Nevertheless, connecting humans together is one of the greatest potentials for VR technology. Although it might seem isolating to put displays between ourselves and the world around us, we can also be brought closer together through successful interaction mechanisms. This section highlights

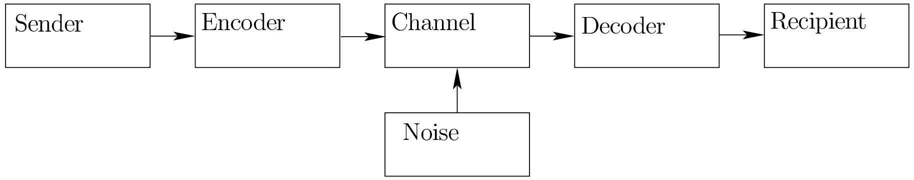


Figure 10.13: The classical Shannon-Weaver model of communication (from 1948). The sender provides a message to the encoder, which transmits the message through a channel corrupted by noise. At the other end, a decoder converts the message into a suitable format for the receiver. This model serves as the basis of communication theory in engineering.

several interesting issues with regard to social interaction, rather than provide a complete review.

Beyond Shannon-Weaver communication An important factor is how many people will be interacting through the medium. Start with a pair of people. One of the most powerful mathematical models ever developed is the *Shannon-Weaver model of communication*, which for decades has been the basis of design for communication systems in engineering; see Figure 10.13. The model involves a *sender* and a *recipient*. The communication system *encodes* a message from the sender, which is then sent over a noisy *channel*. At the other end, the system *decodes* the message and it arrives to the recipient. The recipient could give *feedback* to indicate whether the message has been received intact. This communication model gave rise to the field of *information theory*, which enabled a well-defined notion of *bandwidth* for a communication channel and revealed the limits of data compression.

This model is powerful in that it mathematically quantifies human interaction, but it is also inadequate for covering the kinds of interactions that are possible in VR. By once again following the universal simulation principle, any kind of human interaction that exists in the real world could be brought into VR. The Shannon-Weaver model is inspired by interaction mechanisms such as the 19th century telegraph or 20th century *handheld receiver* (or *walkie-talkie*). In these cases, the humans are completely isolated from each other, and the technology provides a burst of information that is similar to writing a letter. We have gone from text to audio to video communication, and could extend even further by incorporating displays for other senses, such as touch and smell. There are also so many opportunities to use synthetic models, possibly in combination with actual captured information from cameras and microphones. Simple gestures and mannerisms can provide subtle but important components of interaction that are not captured by the classical communication model.

In spite of its shortcomings for VR, keep in mind that the Shannon-Weaver

Figure 10.14: A collection of starter avatars offered by Second Life.

model provides powerful analysis of bandwidth and latency for computer networks and systems, which ultimately support any form of social interaction. Therefore, it has far reaching implications on what can or cannot be accomplished in a VR system. This occurs because all "communication" is converted into streams of bits that are sent through cables or network connections. One key problem is to ensure that the targeted social interaction VR experience is comfortable, convincing, and reliably supported over the computer network.

From avatars to visual capture How should others see you in VR? This is one of the most intriguing questions because it depends on both the social context and on the technological limitations. A clear spectrum of possibilities exists. At one extreme, a user may represent himself through an *avatar*, which is a 3D representation that might not correspond at all to his visible, audible, and behavioral characteristics; see Figure 10.14. At the other extreme, a user might be captured using imaging technology and reproduced in the virtual world with a highly accurate 3D representation; see Figure 10.15. In this case, it may seem as if the person were teleported directly from the real world to the virtual world. Many other possibilities exist along this spectrum, and it is worth considering the tradeoffs.

One major appeal of an avatar is anonymity, which offers the chance to play a different role or exhibit different personality traits in a social setting. In a phenomenon called the *Proteus effect*, it has been observed that a person's behavior changes based on the virtual characteristics of the avatar, which is similar to the way in which people have been known to behave differently when wearing a uniform or costume [33]. The user might want to live a fantasy, or try to see the

Figure 10.15: Holographic communication research from Microsoft in 2016. A 3D representation of a person is extracted in real time and superimposed in the world, as seen through augmented reality glasses (Hololens).

world from a different perspective. For example, people might develop a sense of empathy if they are able to experience the world from an avatar that appears to be different in terms of race, gender, height, weight, age, and so on.

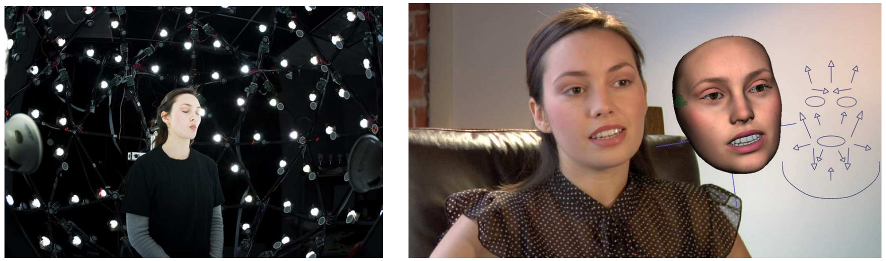
Users may also want to experiment with other forms of embodiment. For example, a group of children might want to inhabit the bodies of animals while talking and moving about. Imagine if you could have people perceive you as if you as an alien, an insect, an automobile, or even as a talking block of cheese. People were delightfully surprised in 1986 when Pixar brought a desk lamp to life in the animated short Luxo Jr. Hollywood movies over the past decades have been filled with animated characters, and we have the opportunity to embody some of them while inhabiting a virtual world!

Now consider moving toward physical realism. Based on the current technology, three major kinds of similarity can be independently considered:

1. **Visual appearance:** How close does the avatar seem to the actual person in terms of visible characteristics?

2. **Auditory appearance:** How much does the sound coming from the avatar match the voice, language, and speech patterns of the person?

3. **Behavioral appearance:** How closely do the avatar's motions match the body language, gait, facial expressions, and other motions of the person?



(a) (b)

Figure 10.16: The Digital Emily project from 2009: (a) A real person is imaged. (b) Geometric models are animated along with sophisticated rendering techniques to produce realistic facial movement.

The first kind of similarity could start to match the person by making a kinematic model in the virtual world (recall Section 9.4) that corresponds in size and mobility to the actual person. Other simple matching such as hair color, skin tone, and eye color could be performed. To further improve realism, texture mapping could be used to map skin and clothes onto the avatar. For example, a picture of the user's face could be texture mapped onto the avatar face. Highly accurate matching might also be made by constructing synthetic models, or combining information from both imaging and synthetic sources. Some of the best synthetic matching performed to date has been by researchers at the USC Institute for Creative Technologies; see Figure 10.16. A frustrating problem, as mentioned in Section 1.1, is the uncanny valley. People often describe computer-generated animation that tends toward human realism as seeing zombies or talking cadavers. Thus, being far from perfectly matched is usually much better than "almost" matched in terms of visual appearance.

For the auditory part, users of Second Life and similar systems have preferred text messaging. This interaction is treated as if they were talking aloud, in the sense that text messages can only be seen by avatars that would have been close enough to hear it at the same distance in the real world. Texting helps to ensure anonymity. Recording and reproducing voice is simple in VR, making it much simpler to match auditory appearance than visual appearance. One must take care to render the audio with proper localization, so that it appears to others to be coming from the mouth of the avatar; see Chapter 11. If desired, anonymity can be easily preserved in spite of audio recording by using real-time voice-changing software (such as MorphVOX or Voxal Voice Changer); this might be preferred to texting in some settings.

Finally, note that the behavioral experience could be matched perfectly, while the avatar has a completely different visual appearance. This is the main motivation for motion capture systems, in which the movements of a real actor are

Figure 10.17: Oculus Social Alpha, which was an application for Samsung Gear VR. Multiple users could meet in a virtual world and socialize. In this case, they are watching a movie together in a theater. Their head movements are provided using head tracking data. They are also able to talk to each other with localized audio.

recorded and then used to animate an avatar in a motion picture. Note that movie production is usually a long, off-line process. Accurate, real-time performance that perfectly matches the visual and behavioral appearance of a person is currently unattainable in low-cost VR systems. Furthermore, capturing the user's face is difficult if part of it is covered by a headset, although some recent progress has been made in this area [18].

On the other hand, current tracking systems can be leveraged to provide accurately matched behavioral appearance in some instances. For example, head tracking can be directly linked to the avatar head so that others can know where the head is turned. Users can also understand head nods or gestures, such as "yes" or "no". Figure 10.17 shows a simple VR experience in which friends can watch a movie together while being represented by avatar heads that are tracked (they can also talk to each other). In some systems, eye tracking could also be used so that users can see where the avatar is looking; however, in some cases, this might enter back into the uncanny valley. If the hands are tracked, which could be done using controllers such as those shown in Figure 10.12, then they can also be brought into the virtual world.

From one-on-one to societies Now consider social interaction on different scales. The vast majority of one-on-one interaction that we have in the real world is with people we know. Likewise, it is the same when interacting through technology, whether through text messaging, phone calls, or video chat. Most of our interaction though technology is targeted in that there is a specific purpose to the engagement. This suggests that VR can be used to take a video chat to the next level, where two people feel like they are face-to-face in a virtual world, or even in a panoramic capture of the real world. Note, however, that in the real world, we may casually interact simply by being in close proximity while engaged in other activities, rather than having a targeted engagement.

One important aspect of one-on-one communication is whether the relationship between the two people is *symmetrical* or *complementary* (from Paul Watzlawick's Axioms of Communication). In a symmetrical relationship the two people are of equal status, whereas in a complementary relationship one person is in a superior position, as in the case of a boss and employee or a parent and a child. This greatly affects the style of interaction, particularly in a targeted activity.

Now consider interactions within a small group of people in the real world. Perhaps a family or coworkers are sharing a meal together. Perhaps children are together on a playground. Perhaps friends and family have gathered for a holiday or birthday celebration. VR versions of such interactions could focus on a targeted activity, such as gathering for a party. Perhaps you are the one who could not attend in person, but will instead "hang out" with the group through some VR interface. Perhaps there is a meeting, and a few people need to attend remotely, which is currently handled by *teleconferencing*, in which voice and video are transmitted over the network. The common scenario that is closest to VR is schoolchildren meeting in a networked video game, with some social interaction occurring while they play. They might form teams and interact through text messaging or voice while playing.

As the number of people increases to over a dozen, the case of a complementary relationship leads to a presentation or interview. Some examples are a teacher lecturing to a class of students, and a politician speaking in front of a group of reporters. In these interactions, a leader has been clearly assigned to communicate with the group. These settings could be reproduced in VR by allowing people to attend through panoramic video capture. Alternatively, the entire event could take place in a virtual world. In the case of a symmetrical relationship, people might mingle at a large reception, and carry on conversations in small groups. This could also be reproduced in VR.

In the limiting case, an online community may emerge, which could connect millions of users. Several examples were given in Section 1.3, including MMORPGs and Second Life. People may have casual interactions by bumping into each other while spending a significant amount of time living or working in a networked virtual world. One issue, which exists in any online community, is membership. Are they open to everyone, or only a closed group?

Transformed social interaction Two common themes in this book have been that VR can produce experiences that are better than reality, and that our per-

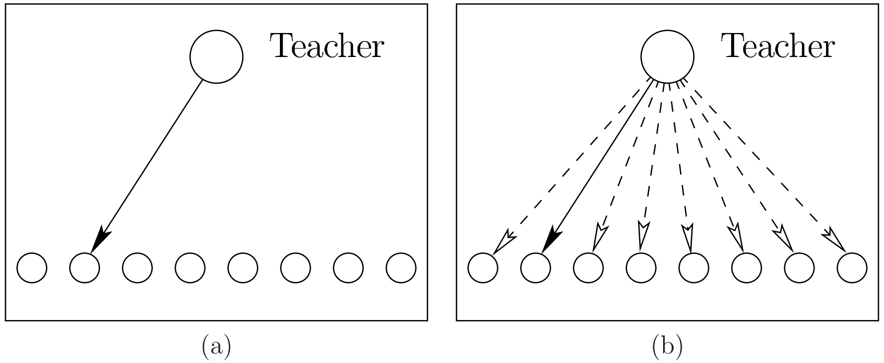


Figure 10.18: (a) A top-down depiction of an ordinary classroom is shown, in which a teacher can look directly at one student. (b) In a VR classroom, the teacher could be looking at each student simultaneously, at least from the perspective of each student.

ceptual systems adapt to new stimuli. It is therefore natural to wonder how social interaction can be altered or improved through VR. The notion of *transformed social interaction* has been introduced Jeremy Bailenson [2]. A thought-provoking example is shown in Figure 10.18. In a virtual world, a teacher could look at every student simultaneously, directly in the eyes, while lecturing to the class. This is physically impossible in the real world, but it is easy to make in VR because each student could see a different version of the virtual world. Of course, the students might reason that the teacher could not possibly be paying attention to *all* of them, but the chance that she *might* be watching could have a significant effect on learning outcomes. The classroom could also appear to have a small number of students, while in reality thousands of students are in attendance. How many more mechanisms for social interaction can be introduced that are impossible to achieve in the real world? How quickly will our brains adapt to them? In what settings would be prefer such interaction to meeting in the real world? The future should bring about many exciting new mechanisms for social interaction.

10.5 Additional Interaction Mechanisms

This chapter has covered three families of interaction mechanisms: locomotion, manipulation, and social. These families emerged from decades of research and development, but do not completely cover every kind of interaction. Many systems demand a custom interaction mechanism be constructed that does not fall into the three families. Furthermore, with the widespread current use of low-cost VR systems, we expect that new families will emerge. A few examples of other interaction mechanisms and associated challenges are presented here.

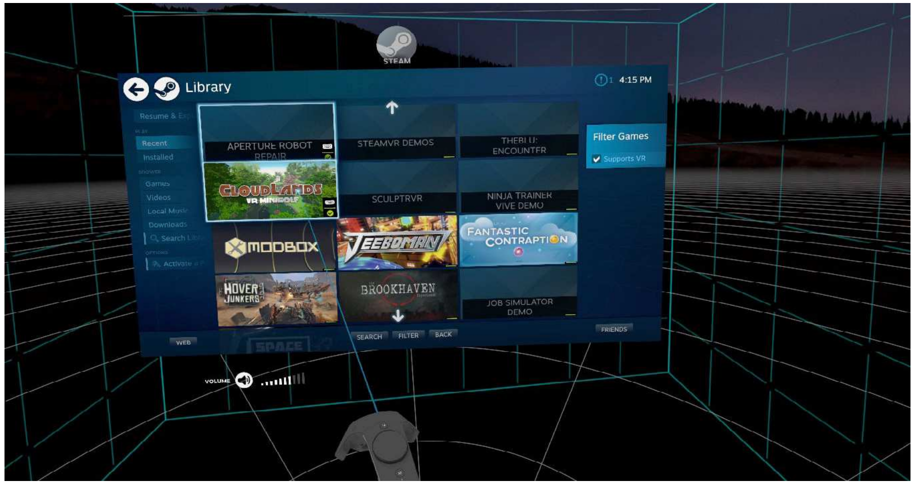


Figure 10.19: The Valve Steam game app store when viewed in the HTC Vive headset.

Interaction with information and media The content of the Internet can be brought into VR in numerous ways by following the universal simulation principle. Figure 1.8 from Section 1.2 showed a movie screen in a virtual movie theater. In this case, simple interaction may be needed to pause or change the movie. As a more complex example, a web browser could appear on a public display in the virtual world or on any other device that is familiar to users in the real world. Alternatively, a virtual screen may float directly in front of the user, while a stable, familiar background is provided; see Figure 10.19.

For decades, people have interacted with the their computers and web browsers using two input devices, one for typing and the other for pointing. In the case of a PC, this has taken the form of a keyboard and mouse. With modern smartphones, people are expected to type on small touch screens, or use alternatives such as voice or swipe-to-type. They use their fingers to point by touching, and additionally zoom with a pair of fingers.

Text entry and editing The typing options on a smartphone are sufficient for entering search terms or typing a brief message, but they are woefully inadequate for writing a novel. For professionals who current sit in front of keyboards to write reports, computer programs, newspaper articles, and so on, what kind of interfaces are needed to entice them to work in VR?

One option is to track a real keyboard and mouse, making them visible VR. Tracking of fingertips may also be needed to provide visual feedback. This enables a system to be developed that magically transforms the desk and surrounding environment into anything. Much like the use of a background image on a desktop

system, a relaxing panoramic image or video could envelop the user while she works. For the actual work part, rather than having one screen in front of the user, a number of screens or windows could appear all around and at different depths.

It is easy to borrow interface concepts from existing desktop windowing systems, but much research remains to design and evaluate completely novel interfaces for improved productivity and comfort while writing. What could word processing look like in VR? What could an integrated development environment (IDE) for writing and debugging software look like? If the keyboard and mouse are replaced by other interfaces, then the user might not even need to sit at a desk to work. One challenge would be to get users to learn a method that offers text entry speeds that are comparable to a using keyboard, but enables them to work more comfortably.

3D design and visualization What are the professional benefits to being able to inhabit a 3D virtual world? In addition to video games, several other fields have motivated the development of computer graphics. Prior to *computer-aided design* (*CAD*), architects and engineers spent many hours with pencil and paper to painstakingly draw accurate lines on paper. The computer has proved to be an indispensable tool for design. Data visualization has been a key use of computers over the past years. Examples are medical, scientific, and market data. With all of these uses, we are still forced to view designs and data sets by manipulating 2D projections on screens.

VR offers the ability to interact with and view 3D versions of a design or data set. This could be from the outside looking in, perhaps at the design of a new kitchen utensil. It could also be from the inside looking out, perhaps at the design of a new kitchen. If the perceptual concepts from Chapter 6 are carefully addressed, then the difference between the designed object or environment and the real one may be less than ever before. Viewing a design in VR can be considered as a kind of *virtual prototyping*, before a physical prototype is constructed. This enables rapid, low-cost advances in product development cycles.

A fundamental challenge to achieving VR-based design and visualization is the interaction mechanism. What will allow an architect, artist, game developer, movie set builder, or engineer to comfortably build 3D worlds over long periods of time? What tools will allow people to manipulate high-dimensional data sets as they project onto a 3D world?

The future Many more forms of interaction can be imagined, even by just applying the universal simulation principle. Video games have already provided many ideas for interaction via a standard game controller. Beyond that, the Nintendo Wii remote has been especially effective in making virtual versions of sports activities such as bowling a ball or swinging a tennis racket. What new interaction mechanisms will be comfortable and effective for VR? If displays are presented to senses other than vision, then even more possibilities emerge. For example, could you give someone a meaningful hug on the other side of the world if they are wearing a suit that applies the appropriate forces to the body?

Further Reading

For overviews of human motor control and learning, see the books [22, 28]. Proprioception issues in the context of VR are covered in [12]. For more on locomotion and wayfinding see [10] and Chapters 6 and 7 of [6]. For grasping issues in robotics, see [23].

For more on locomotion and wayfinding see [10] and Chapters 6 and 7 of [6]. The limits of hand-eye coordination were studied in the following seminal papers: [11, 13, 32]. The *power law of practice* was introduced in [25], which indicates that the logarithm of reaction time reduces linearly with the amount of practice. Research that relates Fitts's law to pointing device operation includes [14, 20, 21, 29]. For broad coverage of human-computer interaction, see [7, 9]. For additional references on social interaction through avatars, see [3, 24, 31].

Bibliography

[1] B. B. Andersen, L. Korbo, and B. Pakkenberg. A quantitative study of the human cerebellum with unbiased stereological techniques. *Journal of Comparative Neurology*, 326(4):549–560, 1992.

[2] J. N. Bailenson, A. C. Beall, J. Loomis, J. Blascovich, and M. Turk. Transformed social interaction: Decoupling representation from behavior and form in collaborative virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 13(4):428–441, 2004.

[3] E. P. Becerra and M. A. Stutts. Ugly duckling by day, super model by night: The influence of body image on the use of virtual worlds. *Journal of Virtual Worlds Research*, 1(2):1–19, 2008.

[4] C. Bergland. The wacky neuroscience of forgetting how to ride a bicycle. *Psychology Today*, May 2015. Posted online.

[5] D. Bowman and L. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings ACM Symposium on Interactive 3D Graphics*, pages 35–38, 1997.

[6] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev. *3D User Interfaces.* Addison-Wesley, Boston, MA, 2005.

[7] P. Cairns and A. L. Cox. *Research Methods for Human-Computer Interaction.* Cambridge University Press, Cambridge, U.K., 2008.

[8] S. K. Card, W. K. English, and B. J. Burr. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics*, 20:601–613, 1978.

[9] J. M. Caroll. *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science.* Morgan Kaufmann, San Francisco, CA, 2003.

[10] R. P. Darken and B. Peterson. Spatial orientation, wayfinding, and representation. In K. S. Hale and K. M. Stanney, editors, *Handbook of Virtual Environments, 2nd Edition*, pages 131–161. CRC Press, Boca Raton, FL, 2015.

[11] J. R. Dejong. The effects of increasing skill on cycle time and its consequences for time standards. *Ergonomics*, 1(1):51–60, 1957.

[12] P. DiZio, J. R. Lackner, and R. K. Champney. Proprioceptive adaptation and aftereffects. In K. S. Hale and K. M. Stanney, editors, *Handbook of Virtual Environments, 2nd Edition*. CRC Press, Boca Raton, FL, 2015.

[13] D. J. Encross. Control of skilled movement. *Psychological Bulletin*, 84:14–29, 1977.

[14] B. W. Epps. Comparison of six cursor control devices based on Fitts' law models. In *Proceedings of the 30th Annual Meeting of the Human Factors Society*, pages 327–331, 1986.

[15] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391, 1956.

[16] A. K. Forsberg, K. Herndon, and R. Zelznik. Aperture based selection for immersive virtual environments. In *Proceedings ACM Symposium on User Interface Software and Technology*, pages 95–96, 1996.

[17] A. Gopnik, A. N. Meltzoff, and P. K. Kuhl. *The Scientist in the Crib: What Early Learning Tells Us About the Mind.* HarperCollins, New York, NY, 2000.

[18] H. Li, L. Trutoiu, K. Olszewski, L. Wei, T. Trutna, P.-L. Hsieh, A. Nicholls, and C. Ma. Facial performance sensing head mounted display. In *Proceedings ACM SIGGRAPH*, 2015.

[19] I. S. MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7(1):91–139, 1992.

[20] I. S. Mackenzie. Movement time prediction in human-computer interfaces. In R. M. Baecker, J. Grudin, W. A. S. Buxton, and S. Greenberg, editors, *Readings in Human-Computer Interaction*, pages 483–492. Morgan Kaufmann, San Francisco, 1995.

[21] I. S. MacKenzie and W. Buxton. Extending Fitts Law to 2D tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 219–226, 1992.

[22] R. Magill and 10th Ed. D. Anderson. *Motor Learning and Control: Concepts and Applications.* McGraw-Hill, New York, NY, 2013.

[23] M. T. Mason. *Mechanics of Robotic Manipulation.* MIT Press, Cambridge, MA, 2001.

[24] P. R. Messinger, E. Stroulia, K. Lyons, M. Bone, R. H. Niu, K. Smirnov, and S. Perelgut. Virtual worldspast, present, and future: New directions in social computing. *Decision Support Systems*, 47(3):204–228, 2009.

[25] A. Newell and P. S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. In J. R. Anderson, editor, *Cognitive skills and their acquisition*, pages 1–55. Erlbaum, Hillsdale, NJ, 1981.

[26] I. Poupyrev, M. Billinghust, S. Weghorst, and T. Ichikawa. The go-go interaction technique: non-linear mapping for direct manipulation in VR. In *Proceedings ACM Symposium on User Interface Software and Technology*, pages 79–80, 1996.

[27] S. Razzaque, Z. Kohn, and M C. Whitton. Redirected walking. In *Proceedings of Eurographics*, pages 289–294, 2001.

[28] D. Rosenbaum. *Human Motor Control, 2nd Ed.* Elsevier, Amsterdam, 2009.

[29] R. W. Soukoreff and I. S. MacKenzie. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts law research in HCI. *International Journal of Human-Computer Studies*, 61:751–759, 2004.

[30] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a WIM: interative worlds in minature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 265–272, 1995.

[31] A. Vasalou and A. Joinson. Me, myself and I: The role of interactional context on self-presentation through avatars. *Computers in Human Behavior*, 25(2):510–520, 2009.

[32] A. T. Welford. *Fundamentals of Skill.* Methuen Publishing, London, 1968.

[33] N. Yee and J. Bailenson. The Proteus effect: The effect of transformed self-representation on behavior. *Human Communication Research*, 33:271–290, 2007.