Chapter 4
VR/AR Input Devices and Tracking

Check for updates

**Paul Grimm, Wolfgang Broll, Rigo Herold, Johannes Hummel,
and Rolf Kruse**

Abstract How do Virtual Reality (VR) and Augmented Reality (AR) systems recognize the actions of users? How does a VR or AR system know where the user is? How can a system track objects in their movement? What are proven input devices for VR and AR that increase immersion in virtual or augmented worlds? What are the technical possibilities and limitations? Based on fundamentals, which explain terms like degrees of freedom, accuracy, repetition rates, latency and calibration, methods are considered that are used for continuous tracking or monitoring of objects. Frequently used input devices are presented and discussed. Finally, examples of special methods such as finger and eye tracking are discussed.

Input devices are used to record user interactions using sensors, as well as other objects and the environment. The data obtained in this way are summarized, if necessary, semantically interpreted and forwarded to the world simulation. There is a wide range of VR/AR input devices available and a classification of these can be done in different ways. The distinction can be made based on accuracy (fine or coarse) or range (from an area that can be reached with an outstretched arm, to an area where one can walk or look around). It is also possible to distinguish between discrete input devices that generate one-time events, such as a mouse button or pinch glove (a glove with contacts on the fingertips) and continuous input devices that generate continuous streams of events (e.g., to continuously transmit the position of a moving object). The physical medium used for determination (e.g., sound waves or electromagnetic fields) can also be used for classification (see Bishop et al. 2001). In the following, the fundamentals of input devices are presented. Then, in Sect. 4.2 tracking techniques are presented in general before a more detailed

---

Dedicated website for additional material: vr-ar-book.org

P. Grimm (✉)
Department of Media, Darmstadt University of Applied Sciences, Darmstadt, Germany
e-mail: springer@paul-grimm.de

107

discussion of camera-based tracking approaches in Sect. 4.3. Sections 4.4 and 4.5 give examples of finger and eye tracking to show how natural user interactions can be detected using specialized input devices. Afterwards (Sect. 4.6) further input devices are presented, which are often used in VR systems. Finally, the chapter is summarized and example questions as well as literature recommendations are given.

4.1 Fundamentals of Input Devices

The interaction of a user with a VR or AR system can be manifold. In a simple case, a conscious action of the user takes place in the form of a push of a button, which is recognized as a unique event by the system in such a way that it can react to it. More difficult are more complex interactions, such as hand movements (e.g., to point at something) or to direct the gaze at something.

This section explains the foundation to describe input devices in more detail. In the case of interactions, a distinction can be made between whether the interaction should be continuous (e.g., in continuous pursuit of a finger pointing at something) or whether part of a movement should be recognized as a gesture (e.g., when point-ing at an object in the virtual world to select it). In both cases, however, the system must be able to track the user, as gestures can only be extracted from recorded data in a subsequent step. It must be determined what exactly is to be tracked by the VR system. Either interaction devices, such as VR controllers or a flystick (see Fig. 4.4), or the user directly can be used. In the latter case, it must then be determined what kind of movements a VR/AR system should detect, or which parts of the body should be considered for interaction (e.g., only the hand, the arm, the head or per-haps the movement of the whole body, as shown in Fig. 4.1 as an example).

Technically speaking, continuous tracking by an input device continuously determines the position and orientation of an object (e.g., hand or head, controller). This process is called tracking. For simplification, an object is usually regarded as a so-called rigid body that cannot be deformed.

The movement of a rigid body can be broken down into a displacement (transla-tion) in space and a rotation around three perpendicular axes. Thus, the movement of a rigid body can be specified by giving six values (three coordinates as position and three angles to describe the orientation) for each time step. These independent move-ment possibilities are called degrees of freedom. Generally, a system with N points has $3 \times N$ degrees of freedom (each point in space has three degrees of freedom, N points in space corresponding to $3 \times N$ degrees of freedom), which in turn are reduced by the number of constraints. In the case of rigid bodies, where all distances between points are constant, there are always six degrees of freedom left (Goldstein 1980). As an example, a cube can be used that has eight vertices and thus $3 \times 8 = 24$ degrees of freedom. If the cube is considered to be non-deformable, the constraints are that the respective distances between the eight points remain unchanged. For eight points, this means $6 + 5 + 4 + 2 + 1 = 18$ constraints ($4 + 2$ for the distances including the diagonals of the flat base surface, $3 + 2$ for the first side surface includ-ing the diagonals, two for the next side surface and one for the last side surface).

> Degrees of Freedom (DOF) are the independent movement possibilities of a
> physical system. A rigid body has six degrees of freedom: three each for trans-
> lation and rotation.

The goal of tracking is to determine or estimate the values corresponding to these
six degrees of freedom (6DOF) of the tracked objects for continuous interaction.
The data acquisition is usually performed in the reference system of the respective
tracking system. If several or even different systems are used, the tracking data must
be transferred to a common reference system.

Starting with mechanical tracking systems (see Sect. 4.6.2), through the use of strain gauges to camera-based approaches (see Sect. 4.3), data was recorded in different ways, as was data transmission by cable or radio. Correspondingly, very different input devices are available, which have different advantages and disadvantages. Input devices can be described by the following characteristics.

Number of Degrees of Freedom Per Tracked Object
The number of specific degrees of freedom per tracked object varies depending on the input device. Usually, the determination of all six degrees of freedom by an input device is desirable. However, it also happens that only the position – equivalent to the three degrees of freedom of translation – or only the orientation – equivalent to the three degrees of freedom of rotation – is determined. Examples of the limited determination of degrees of freedom are the compass (one degree of freedom, determination of the orientation in the plane) and GPS, which, depending on the number of visible satellites, determines two to three degrees of freedom of translation. It is also possible that the accuracy of the determination of individual degrees of freedom is different (in the case of GPS, the position on the Earth's surface is recorded more accurately than the height above it).

Number of Objects Tracked Simultaneously
Depending on the application, it is important to consider how many objects are to be tracked simultaneously. In addition to tracking the user or recording the viewer's point of view, other objects (e.g., one or more input devices) often need to be tracked. For the use of several objects it is helpful if they can not only be tracked, but they can be uniquely identified by an ID. It is helpful if these IDs are retained, even if individual objects are temporarily out of monitoring.

Size of the Monitored Area or Volume
The size of the monitored area or volume varies greatly depending on the type of input device used. It must be ensured that the selected input devices offer an area that is large enough for the requirements.

Depending on the application, this can mean that it is sufficient to cover an area that can be reached with the arm or that corresponds to the movements of a head in front of the monitor. There are also applications where it is necessary to be able to walk around. The reason for the size restrictions may be that the input device is wired, has a mechanical construction or (in the case of camera-based input devices) the resolution is too low. Depending on the technology used, the shape of the monitored area may vary (e.g., similar to a circle in the case of wired technologies or similar to a truncated pyramid in the case of camera-based technologies with one camera).

Accuracy
Not only because of the physical limitations of the input devices, high accuracy is not always achievable. Sometimes it is also a question of cost. For example, in optical tracking a change of camera can increase the accuracy. However, if an expensive industrial camera is used instead of a simple webcam, the price can easily increase by a factor of 10 or more. Depending on the application, it must be considered what accuracy is necessary or what budget is available. The usual range in spatial

resolution is between millimeter accuracy (e.g., optical finger tracking) up to an inaccuracy of several meters (e.g., when using GPS). The accuracy can also vary with different types of degrees of freedom (translation or rotation), e.g., as in the case of GPS, where altitude determination is not as accurate as position determination. The accuracy can also be position-dependent: for example, the accuracy may be lower at the edge of the monitored area than at its center. During digitization, the measured values are quantized, e.g., to 8 bits or 16 bits. With regard to the measurement technology, noise (addition of an interfering signal), jitter (temporal inaccuracy of the time of measurement or of the sampling time) or interpolation errors can also be assumed to be interfering influences.

Update Rate
The update rate describes the resolution of an input device in time. The degrees of freedom are determined in discrete time steps. The number of these measurement points per second is called the update rate. Thus, monitoring the real continuous motion of an object (shown as a black line in Fig. 4.2) results in corresponding measuring points. Basically, a time-discrete signal is obtained, which will usually have errors. Figure 4.2 shows some of the possible errors.

Latency
Each input device requires a certain amount of time to react (e.g., time until the next scan, due to signal propagation times in cables or due to the processing of data by algorithms), which causes a delay. This is called latency. An example of the effect can be seen in Fig. 4.2. The significance of latency for VR systems is discussed in more detail in Sect. 7.1.
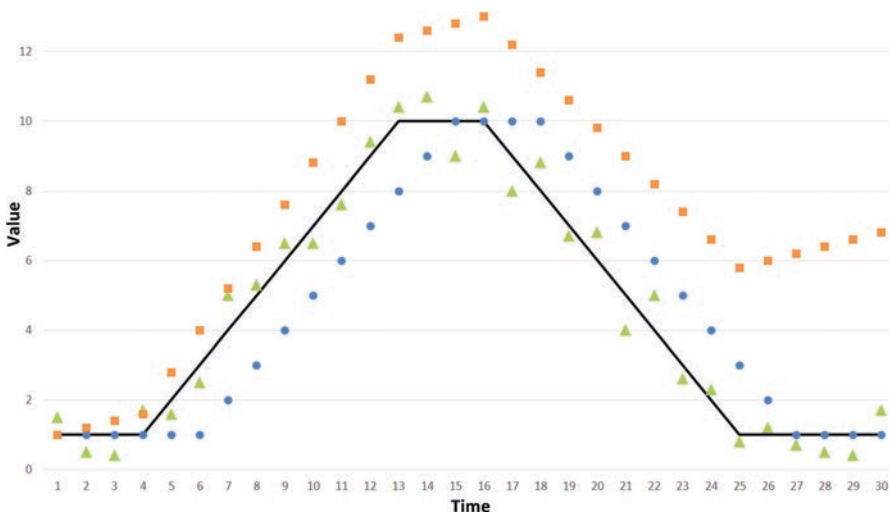

Fig. 4.2 Possible errors during data acquisition of the position of a moving object (black line): acquisition with latency (blue dots), with drift (orange squares) and with noise (green triangles), displayed over time (horizontal axis)

Drift

Errors that keep adding up can cause drift. If input devices record relative changes (e.g., change in position compared to the previous scanning or the previous measuring point), errors can increase over time. An example of drift is shown in Fig. 4.2.

Sensitivity to External Conditions

Depending on the technology used, the external conditions must be observed. Lighting or temperature can have just as much influence as the furnishing of the room in which the VR system is set up. Uniform lighting can be of great advantage, especially with optical methods, compared with hard transitions from direct sunshine to shaded areas. It would be annoying not to be able to use a tested application because the sun appeared from behind a cloud. A problem has often been reported to have arisen in trade fair construction, where before the opening usually only some working lights were used, but during trade fair operations there were often many other spotlights, which then led to disturbing influences.

With optical tracking systems it can be helpful to work in darkened rooms and to create the desired lighting situation with artificial light. It should be noted that direct light sources can interfere with camera sensors. Methods based on sound are often susceptible to different temperatures or different air pressures, as this changes the speed of sound (on which the measurement is based). Electromagnetic methods in turn react sensitively to (ferro-)magnetic materials and electromagnetic fields in the rooms (e.g., metallic table frames or the power supplies of other devices).

Calibration

Calibration is the adjustment of measured values to a given model. For both virtual reality and augmented reality, the measured values must be adjusted to the real objects used, so that the real movements that are tracked also correspond to the dimensions in the virtual world. With optical methods, this also includes the determination of imaging errors of the optics (e.g., distortions).

Usability

For the application it can be decisive to what extent a user is restricted by the input devices. For example, it may be necessary to put on glasses or shoes or hold VR controllers. It also makes a difference for the application whether the respective devices are wired or connected via radio technologies. The size of the room in which a user is allowed to interact also influences whether the user can immerse himself or herself in the application or whether he or she must constantly ensure that he or she does not exceed predetermined interaction areas. It may also be necessary that the user is always oriented towards the output device to enable good tracking. A detailed consideration of usability is given in the framework of the consideration of basics from the field of human–computer interaction in Sect. 6.1.

The obtrusiveness of an input device can be seen as a measure of the extent to which it is considered to be disruptive. For example, it makes a big difference whether a head-mounted display can be worn like sunglasses or whether it can be used like a bicycle helmet due to its weight and dimensions.

4.2   Tracking Techniques

As explained in the introduction, tracking is the continuous estimation of the position and orientation of an object. Generally, we may distinguish between systems in which the measuring sensors are located on the tracked objects themselves and determine their position and location in relation to their surroundings (inside-out tracking), and systems in which the measuring sensors are distributed in the environment and interact to measure an object from outside (outside-in tracking) (see Sect. 4.3 on camera-based tracking). The determination or estimation of the position of an object is carried out in a defined coordinate system. One possibility is the estimation in relation to individual objects. Here, the relative transformation between the user or camera coordinate system and the object coordinate system is determined for each object. Another possibility is that several objects use a common coordinate system. In this case, the transformations between the individual objects within the coordinate system must be known, and the transformation between the camera and this coordinate system is estimated. If only the position of some objects in a global coordinate system is known, while others can change their position and orientation within it, you get mixed forms of both scenarios.

In the following, different tracking techniques are presented with their advantages and disadvantages. Camera-based tracking techniques will be presented in Sect. 4.3 due to their diversity.

4.2.1   Acoustic Tracking

Acoustic-based input devices use the differences in the time of flight (TOF) or phase of sound waves. Ultrasound that is inaudible to humans (sound waves with a frequency of more than 20,000 Hz) is used. The measurement uses a transmitter and a receiver, where one of them is connected to the tracked object. This allows for the determination of the distance between them. By that, the position of an object can be limited to a spherical surface around the transmitter. By adding a second transmitter or a second receiver, the position can already be limited to a circular path (as an intersection of two spheres). Adding a third transmitter or receiver then limits the position to two points (as an intersection of three spheres or as an intersection of two circles). A plausibility check is then used to determine the actual position from these two points. A setup with one transmitter and three receivers (or three transmitters and one microphone) thus allows for the determination of all three degrees of freedom of the translation (3 DOF). If the orientation is also to be determined (6 DOF), three transmitters and three receivers must be used.

Compared to other 3D tracking systems, acoustic systems are rather cheap. A disadvantage of acoustic tracking is its sensitivity to changes in temperature or air pressure. Any change in temperature or air pressure requires a (re)calibration of the system.

4.2.2 Magnetic Field-Based Tracking

Magnetic fields can be used for tracking. However, a distinction must be made between artificial magnetic fields and the Earth's magnetic field. In mobile systems, so-called fluxgate magnetometers (also known as Förster probes) are usually used for electronic measurement of the Earth's magnetic field. Based on the individual sensor orientation, both the horizontal and vertical components are measured. This gives two degrees of freedom of the current position. Sensors for magnetic field measurement are disturbed easily by artificial magnetic fields in their environment. Especially indoors, electromagnetic fields (e.g., from installed cables) can falsify the recorded data to such an extent that they become useless for determining the position. In smartphones and tablets, three orthogonal magnetometers are usually combined with three linear inertial sensors and three angular rate sensors each (cf. Section 4.2.3) to compensate for measurement errors through redundancy.

For indoor systems, the use of the Earth's magnetic field is usually not possible due to disturbing influences. However, with the help of current-carrying coils, artificial magnetic fields can be created which can then be used for tracking. Coils are also used as sensors. Depending on whether a static magnetic field (direct current, DC) or a dynamic magnetic field (alternating current, AC) is used for the measurement, different measuring methods are used. With alternating magnetic fields, the magnetic field induces currents in the coils, which are used as a measure of the position and orientation in the magnetic field (or in space). In DC magnetic fields, a current flow through the receiver coils and a voltage drop can be observed perpendicular to both the direction of current flow and the magnetic field direction when the coils are moved through the magnetic field. This so-called Hall effect also allows tracking by measuring the Hall voltage. The combination of three orthogonal transmitters and three orthogonal receiving coils allows one to determine the position and orientation in space. The advantages of electromagnetic tracking systems are that the receivers are small and that they are insensitive to occlusion by the user or other non-conductive objects. The major disadvantage is that no (ferro-)magnetic materials must be used in the room (up to the use of plastic screws for fastening the sensors) and no electromagnetic fields should exist, as these interfere with the magnetic field, introducing measurement errors. Since interference influences, especially in a room with other electromagnetic components of a VR or AR system, can usually not be avoided, complex calibration procedures are necessary to compensate for disturbing interference. However, this assumes that the interference is exclusively based on static, permanently mounted objects.

4.2.3 Inertial Tracking

Inertial tracking is based on sensors that measure acceleration (called inertial sensors or acceleration sensors). Inertial tracking is primarily used to determine orientation. One area of application is, among others, the detection of the joint positions of a user by attaching appropriate sensors to the individual limbs.

Depending on the design, a distinction is made between linear inertial sensors, which measure the acceleration along an axis, and angular rate sensors, which measure the angular acceleration around an axis. Since the latter behave like a gyrocompass (gyroscope), they are sometimes also called gyro sensors. Together they form a so-called Inertial Navigation System (INS). Typically, three linear inertial sensors (translation sensors) and three angular rate sensors, arranged orthogonally to each other, are integrated into an inertial measurement unit (IMU). Such units often also include three magnetometers, which are also arranged orthogonally to each other (see Sect. 4.2.2).

Linear accelerometers can be used to determine the orientation, but only in the idle state. Then, the inclination to the vertical can be measured due to the direction of gravity. Since the orientation in the horizontal is perpendicular to gravity, this cannot be measured by linear inertial sensors. For input devices that can be moved freely, three orthogonal sensors are nevertheless installed so that at least two can be used for measurement at any time. However, linear inertial sensors may also be used for position determination. Based on the linear acceleration values in the three orthogonal sensors, the current speed can be estimated by integration and the change in position by a second integration. However, due to measurement inaccuracies (usually amplified by a relatively low accuracy in converting the analog measured values into digital values), drift effects often occur. This means that if, for example, a sensor is moved out of its resting state and then stopped again, the sums of the recorded acceleration values would have to add up to zero at the end, resulting also in zero velocity. However, this is usually not the case, so that the measurement results in a low residual speed even in the idle state. This leads to an increasing deviation between the measured and actual positions over time.

In the case of acceleration sensors for measuring angular velocity, the acceleration values are integrated twice analogously to obtain the angle of rotation. This also causes the problem of drift. It is therefore recommended to recalibrate in the idle state using the linear accelerometers. For the detection of rotations over all three axes, three sensors are usually installed orthogonally to each other, even with gyro sensors.

4.2.4   Laser-Based Tracking

In laser-based tracking, the tracked objects are equipped with several photosensors that detect laser beams emitted from a base by two rotating lasers. If only one base is used, the photosensors are often occluded, e.g., by the user's own body. Most systems therefore use several base stations. This also allows a larger tracking volume to be covered. For synchronization between the base stations and the objects, either additional infrared signals are used, or the sync signal is transmitted via the laser beam itself. The lasers rotate around a horizontal or vertical axis, whereby the laser beam is emitted only in one direction with a certain aperture angle (e.g., 120°). The position and orientation of the object can be calculated based on the time

difference between the detection of the laser light by the individual photosensors. At a defined rotation speed of the lasers (e.g., 1000 Hz), the position is determined by the time difference between the infrared flash, which is emitted before the start of a laser rotation, and the impact on one of the sensors. At a rotation frequency of 1000 Hz, an aperture angle of 120° and a time difference of 1/6 ms from the infrared synchronizing flash, this results in a position at the center of the monitored space.

4.2.5   Outdoor Position Tracking

In the field of mobile outdoor applications, global satellite-based systems such as GPS, Glonass or Galileo are used for positioning. Mobile position tracking is especially relevant for AR, since VR applications are typically not used outdoors. However, in contrast to navigation applications, where satellite data can be compared with existing roads and paths, the position of an AR system is almost arbitrary. Thus, deviations of 10 m and more are not uncommon. Especially under poor reception conditions, the accuracy can be reduced even further. Global satellite-based systems usually require a view of at least four satellites to determine their position. While this is usually not a problem outdoors, reception inside buildings with conventional receivers is not suitable for AR. But even in forests and deep valleys the reception quality can be significantly impaired, so that positioning is not possible or only possible to a limited extent. A particular problem is the use in inner city areas. Due to high buildings and narrow alleys, the free view of the satellites may be so limited that proper positioning cannot always be guaranteed. Here, one also speaks of 'urban canyons' (see Fig. 4.3).

While conventional GPS signals are not sufficient for AR in most cases, the accuracy can be significantly increased by using differential methods. A distinction is made between Differential GPS (DGPS) and Satellite Based Augmentation System (SBAS). With DGPS, a correction signal is calculated based on a local reference receiver whose position is known. This correction (received by radio or via the Internet) is then applied to the locally received GPS signal, allowing accuracies down to a few centimeters. In SBAS, the reference system is formed by several geostationary satellites. These reference satellites each provide correction data for specific areas (WAAS in North America or EGNOS in Europe). Based on SBAS, accuracies of about one meter can be achieved. However, SBAS (in particular) in city centers is again sometimes problematic due to the often limited visibility to the south (geostationary satellites have an orbit above the equator). For outdoor AR applications, however, the use of SBAS is usually the only way to achieve an acceptable positioning accuracy. This is already sufficient for the augmentation of objects and buildings that are not in the immediate vicinity of the observer. If DGPS is used, augmentation can usually be achieved even at a short distance without any noticeable deviation from the actual position. However, the objectively perceived quality of the positioning strongly depends on whether the virtual object must fit seamlessly

Fig. 4.3 Buildings block
GPS signals in so-called
urban canyons

to a real object or can be positioned rather freely (for example, a virtual fountain on a real site).

In addition to DGPS and SBAS, Assisted GPS (A-GPS) and WLAN positioning are also frequently used, especially in smartphones and tablets. With A-GPS, an approximate position is determined on the basis of the current mobile radio cell (possibly refined by measuring the signal propagation times to neighboring mobile radio masts), whereas WLAN positioning uses known WLAN networks (these do not have to be open, but only uniquely identifiable). Neither method provides sufficiently accurate position data for AR. However, A-GPS can also significantly accelerate the start-up phase of an ordinary GPS receiver by transmitting satellite information (especially current orbit data and correction data). This is particularly relevant for AR applications if the users are frequently in areas where there is no satellite reception – for example in buildings.

4.3   Camera-Based Tracking

In recent years, camera-based tracking, also known as optical tracking, has become increasingly popular because it enables high accuracy and flexible use. In the field of optical tracking different techniques are used. They are based on the idea of using objects recorded in the video stream to determine the relative positioning and orientation of the objects to the camera (the so-called extrinsic camera parameters) (Hartley and Zisserman 2000).

Basically, techniques can be distinguished according to whether markers (see Fig. 4.8) are used for tracking which are easily recognizable in the recorded video stream (by their color, shape, contrast, brightness, reflective properties, etc.), or whether the method also works without markers (markerless). In the latter case, either lasers are used, or cameras capture features within the camera image (see Sect. 4.3.3). It is also possible to distinguish between methods in which the cameras are directed at the object to be monitored from the outside (outside-in), or whether the cameras are mounted to the object to be monitored and record the surroundings (inside-out). In most cases, outside-in methods combine several cameras with the aim of increasing the area of interaction or making it less susceptible to occlusion. The disadvantage of outside-in methods is that a (very) large number of cameras may be required to monitor large interaction areas and that the overall costs may rise rapidly, especially when using special cameras. The disadvantage of inside-out procedures is that the user must accept restrictions by carrying cameras around. Even though camera modules have become very small nowadays, the total package of camera and possibly battery and transmission or evaluation logic is relatively heavy. The advantage is that users are not restricted to a certain interaction space and can therefore move around more freely.

From the user's point of view, a markerless outside-in method would of course be desirable, as this is where the restrictions for the user are least severe. Users do not have to hold anything in their hands, do not need markers (e.g., on clothing) and can move freely and walk freely through the room. In practice, however, it has been shown that markerless tracking systems are more susceptible to interference (e.g., additional people in the room or changing lighting conditions) than marker-based systems, and that the accuracy of marker-based systems is often higher.

4.3.1   Marker-Based Methods

To reduce the complexity of calculations and to avoid errors in different lighting situations, optical tracking techniques often use clearly specified markers whose image can be quickly identified in the video stream via threshold filters. Basically, active and passive marker can be distinguished, depending on whether the markers passively reflect the light or they themselves actively radiate light. Figure 4.4 (top) shows an example of a six degrees of freedom controller with active markers (18

Fig. 4.4 (Top) VR controller with active marker; (bottom) **c**ameras with infrared LEDs for illumination and flysticks with reflective markers

white LEDs arranged in a given pattern). Figure 4.16 shows a similar controller with active infrared LEDs.

When using RGB cameras, black and white markers with defined sizes are often used for this purpose. These are discussed in detail in Sect. 4.3.2. There are also different approaches with colored markers. However, due to the lighting situation and possibly also due to inferior cameras, even areas that are actually monochrome are usually no longer monochrome in the video stream, so that the susceptibility to errors increases when searching for a colored area. Better results can be achieved using color-based tracking with active markers, i.e., self-luminous markers. Electric lights (with the disadvantage of the power supply) such as the PlayStation Move controller or glow sticks (also known as bend lights, which use chemiluminescence) have proven to be very useful for this purpose.

To allow illumination of a scene without dazzling the users, infrared cameras are often used in VR. The markers used here are either passive reflectors in combination with infrared lights or active infrared LEDs such as the Nintendo Wii (see Lee 2008). Figure 4.4 (bottom) shows the infrared LEDs used for illumination. In the video stream, small very bright round areas can be seen for each marker. The

visibility of a marker in several camera views allows the three-dimensional position to be calculated.

Single markers are sufficient if tracking is only to provide the position (3 DOF). However, a rigid body (also called a target in some tracking systems) typically requires the calculation of its position and orientation. Consequently, a target is composed of several individual markers. In a calibration step, the geometric structure of the targets (e.g., the distances of the individual reflection spheres) must be communicated to the tracking system. If all targets differ in their geometric structure, identification can be made based on these characteristics. In Fig. 4.4 (right side) two input devices with targets are shown, which take over the function of a 3D joystick, and with which the user can indicate positions and orientations in 3D space (so-called flysticks).

To make the reflection of passive markers as efficient as possible, retroreflection is usually used. Retroreflection means that the beams of light are reflected specifically in the direction of the incident light and is based on two basic optical principles: in the case of reflection by triple mirrors, the mirrors are arranged with a right angle in between, as shown in Fig. 4.5 (left). When reflected by glass spheres, the spheres focus the incoming light approximately on the opposite surface of the glass sphere (see Fig. 4.5, right). A layer of microscopically small glass spheres applied to reflective material acts as a retroreflector. These foils can be produced on flexible carrier material and are therefore used to produce ball markers as shown in Fig. 4.6 and Fig. 4.7.
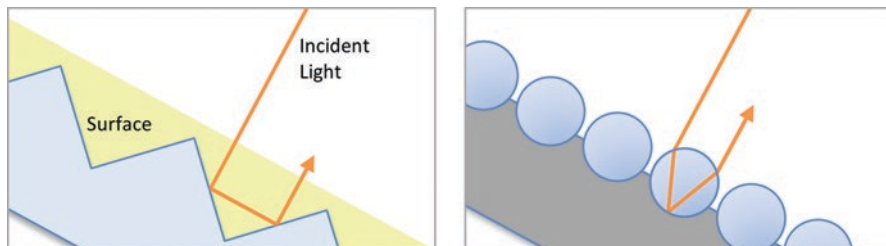

Fig. 4.5 Retroreflection of protected triple mirrors and glass spheres.

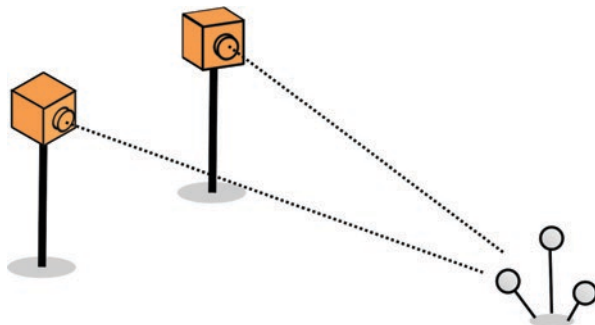**Fig. 4.6** Tracking a target from two cameras

Fig. 4.7 Optical tracking of a person with reflective markers (the markers appear to be illuminated by the flashlight used) and several infrared cameras (infrared LEDs appear red)

Active markers often use infrared LEDs that must be synchronized with the cameras. This synchronization can be done with active markers via an IR flash. The cameras emit IR flashes that are reflected by the markers towards the camera lens. Due to the IR flashes, it is possible that opposite cameras are blinded. A common solution for this is to divide the cameras into so-called flash groups that work alternately, so that the opposite camera is inactive when taking the picture.

The tracking cameras that scan a specific area register the reflected radiation in a grayscale image. The pre-processing of this image data takes place in the camera and provides 2D marker positions with high accuracy using pattern recognition algorithms optimized for circular surface detection. To be able to determine the coordinates of a marker or target in space at all, it is necessary that at least two cameras scan the same area simultaneously (cf. Figure 4.6). Larger volumes are accordingly built up with more cameras, whereby it must also be ensured that partial areas of overlap are scanned by additional cameras. It is therefore important to ensure that the individual areas are linked.

The calibration of outside-in procedures with markers is usually carried out with the aid of test objects known in shape and size, which are moved in the monitored room. The test data obtained in this way allows the coordinate systems of the individual cameras to be aligned with each other such that tracked objects can be described in a uniform coordinate system.

The camera 2D data is transmitted to the central tracking controller, which calculates the 3D positions of the marker or the 6D data of the rigid bodies by triangulation and passes them on to the user. To enable the tracking software to perform this triangulation, the exact positions and orientations of the tracking cameras must be known. In a typical VR system, the accuracy requirement for this is less than 1 mm in position and less than 0.1° in angle. To determine the position and orientation of the tracking cameras with this precision, the tracking software provides a simple calibration step whose basic mathematics (bundle adjustment) is derived from photogrammetry (Hartley and Zisserman 2000) and which allows the calibration in a short time. To achieve a coverage of the tracking volume according to the

requirements, the tracking cameras are equipped with lenses of different focal lengths. This allows a variation in the field of view (FOV). To allow unrestricted working in front of power walls or in multi-side projections, wide-angle lenses for the tracking cameras are selected. It is important that the user can get close to the projection screens to achieve high immersion. Figure 4.7 shows an example where an optical tracking system is used to capture the movement of a user, so-called motion capturing.

Optical tracking in closed multi-sided projections (such as 5- or 6-sided CAVEs; see Sect. 5.4.2) presents a special problem. Optical tracking through projection screens is not possible because these screens have a highly scattering surface and optical imaging through a scattering surface is generally difficult. Therefore, tracking cameras must be installed inside the CAVE, which leads to an impairment of the spatial impression in the virtual environment by these camera bodies. For multi-sided projections in particular there are special cameras that are installed in the corners of the multi-sided projection, looking through a hole of about 40 mm diameter. This allows precise optical tracking in CAVEs to be used, whereby the optical interference caused by the holes in the corners is negligible according to the users.

4.3.2 Tracking Using Black and White Markers

Camera-based tracking using markers has been used for AR since the late 1990s and the technique is still in use today. In most cases, markers with black and white patterns are used (see Fig. 4.8). Compared to colored markers, these offer the advantage that they can be extracted from images with the aid of simple threshold values, even under varying brightness conditions.

The markers used are usually either square or round and bordered by a completely black or completely white border. Criteria for selecting one of the systems can be stability, recognition speed or the number of distinguishable marks. Some of the better-known marker-based tracking approaches include ARToolkit, ARTag, ARToolkit+ or the IS 1200 VisTracker. For a detailed comparison between different marker-based approaches, see Köhler et al. (2010).

Fig. 4.8 Typical markers as used for camera-based tracking

Use of Marker Tracking

For marker tracking, the pattern and size of the individual marker must be known in advance. While some methods (such as ARToolkit, cf. Berry et al. 2002) allow any black and white patterns for the inner part of the marker, the possible patterns are predefined in other methods (such as ARToolkit+). The latter prevents performance losses with many markers. As a rule, markers must be completely visible in the captured camera image to be recognized. With predefined patterns, however, redundancy can often still be used to detect a marker that is not completely visible. If markers are too large, it can also happen that only a part of the marker is visible when the camera is very close to it and tracking is therefore not possible or only possible to a limited extent. Conversely, if the marker is too small in the camera image, this leads to both faulty pattern recognition due to the too small number of detected marker pixels and to a significant reduction of the tracking accuracy, such that even with static objects and a practically motionless camera, transformation values can vary greatly. In addition to the size of the marker, the resolution of the camera is a decisive factor. If the AR application requires that users look at an object from very different distances, it can be advantageous to use markers of different sizes in parallel. A universal solution for this problem is the use of fractal markers (Herout et al. 2012). In addition to the distance, the angle between camera and marker as well as the current lighting situation have a major impact on the quality of the tracking results. If the angle becomes too flat, the calculated transformation values often start to vary greatly (Abawi et al. 2004). If the lighting is too bright (also due to reflections) or too dark (also due to shadows), white and black marker areas are ultimately no longer recognized sufficiently clearly from each other, making tracking no longer possible.

The main advantages of marker-based tracking are that the markers can be created quickly and easily by printing them out and can be applied to objects, walls and ceilings, or can be easily integrated into books and magazines. Even though AR markers may look similar in parts, they should not be confused with QR codes, which are used to encode strings of characters, especially URLs.

The main disadvantage of markers is that they usually must be applied directly to or on the object to be augmented. This is due to the fact that the markers would otherwise often not be visible when looking at the object (more closely) as well as because tracking inaccuracies have a much stronger effect on augmented objects if the distance from the marker to an augmented object get bigger. The markers are therefore often disturbing with respect to the real object. Another aspect is that it is not possible or not appropriate to place markers on many real objects (for example on a statue). An aggravating factor for smaller objects is that when interacting with the object (for example, by touching it), the markers are easily covered by the user's hand or arm, either completely or partially, so that tracking is no longer possible. There are numerous other factors that influence the quality of tracking. An essential aspect is the quality of the camera and the camera calibration (see Szeliski 2011). Another problem is that with some methods (such as ARToolkit) the performance decreases reciprocally quadratic with the number of patterns to be detected.

Basic Operation

In the following, the basic procedure of marker-based tracking will be outlined using ARToolkit (Kato and Billinghurst 1999) as an example. The tracking is basically done in four steps:

1. Camera captures video image
2. In the picture, the system searches for areas with four connected line segments
3. It is checked whether the detected areas represent one of the predefined markers
4. If a marker was found, the position and orientation of the camera to the marker are calculated from the position of the vertices in the image

After obtaining the current camera image, it is first converted to a grayscale image. A black and white image is then generated based on a threshold value, whereby all values below the threshold value are displayed in black and those above the threshold value in white. All line segments in the image are now identified and then all contours are extracted from line segments with four lines. The parameters of the line segments and the positions of the corner points are temporarily stored for later calculation (see Fig. 4.9).

The region found within the four vertices is then normalized. As the surrounding black border has a uniform width of 25% of the edge length, the image to be compared can be easily extracted from the center of the image. The image is then tested for matching with the stored patterns (see Fig. 4.10). For the comparison of each stored pattern, the four possible orientations at three brightness levels each are used. The pattern with the highest degree of similarity is recognized if a defined threshold value for similarity is exceeded. It is therefore also important to select patterns with the lowest possible similarity between them to avoid false positives. Based on the orientation of the pattern, the recognized vertices can easily be assigned to the corresponding coordinates in the marker's coordinate system.
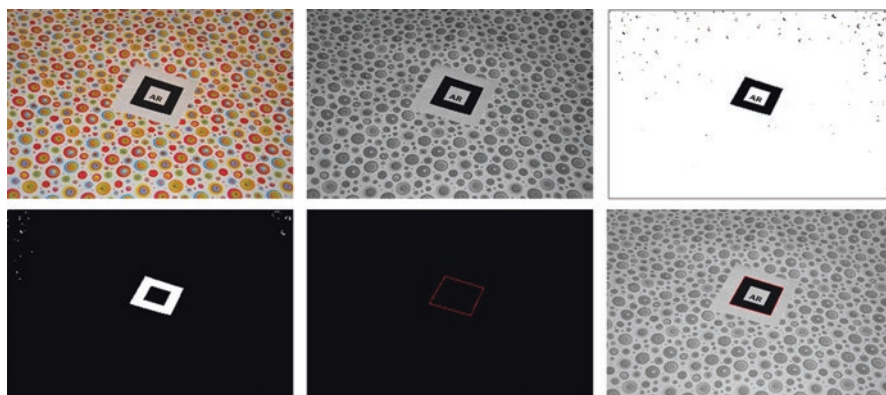

Fig. 4.9 Single steps in the recognition of the marker boundaries in the camera image: conversion to grayscale image, black and white image based on a threshold value, segmentation, identification of lines, identification of contours from four lines and storage of the corner points

Fig. 4.10 Recognized marker, normalized marker, normalized original

Intrinsic and Extrinsic Camera Parameters

The calculation of the pose of the marker in relation to the camera is based on the mapping of the marker's corner point coordinates to pixels. The size of the marker must be known.

T_{cm} is the transformation matrix from the marker coordinate system M to the camera coordinate system C. The position of the camera corresponds to the optical center and thus the origin of the camera coordinate system. The viewing direction of the camera is along the negative z-axis of this coordinate system. $\vec{v}_m$ is a coordinate in the marker coordinate system M and $\vec{v}_c$ the coordinate transformed into the camera coordinate system C. For a detailed representation of the relationships see Fig. 4.11. Thus, the following applies:

$$\vec{v}_c = T_{cm} \cdot \vec{v}_m$$

and

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_m \\ y_m \\ z_m \\ 1 \end{bmatrix}$$

wherein the homogenous matrix T_{cm} is composed of a 3×3 rotation matrix R and a translation vector $\vec{t}$. Both components have three degrees of freedom each; the whole transformation thus has six. Camera calibration (cf. Szeliski 2011) yields the intrinsic camera parameters and thus the calibration matrix K, which determines the mapping of the camera coordinates to the image plane S. Here applies:

$$K = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where f is the focal length of the camera (distance from the image plane) and (c_x, c_y) is the optical center of the image in image coordinates. Strictly speaking, this is an idealized (pinhole) camera, where it is assumed that the focal length is the same in both sensor dimensions and that there is no distortion due to a non-perpendicular installation of the camera sensor (cf. Szeliski 2011, p. 47). Thus, the relation between a camera coordinate $\vec{v}_c$ and an image pixel $\vec{v}_s$ can be described by

$$\vec{v}_s = \begin{bmatrix} s_x \\ s_y \\ s_z \\ s_w \end{bmatrix} = \begin{bmatrix} K & 0 \\ 0 & 1 \end{bmatrix} \cdot \vec{v}_c$$

where $\vec{v}_s$ the must be normalized so that $s_z = 1$ (Fig. 4.11).

By inserting the detected pixels and using the calibration matrix K and the known distance between the vertices, and taking into account the orientation known from the marker orientation, the 3×3 rotation matrix R and the translation vector of T_{cm} can thus be determined. These are called extrinsic camera parameters. For further details of the method see Kato and Billinghurst (1999) and Schmalstieg and Höllerer (2016).
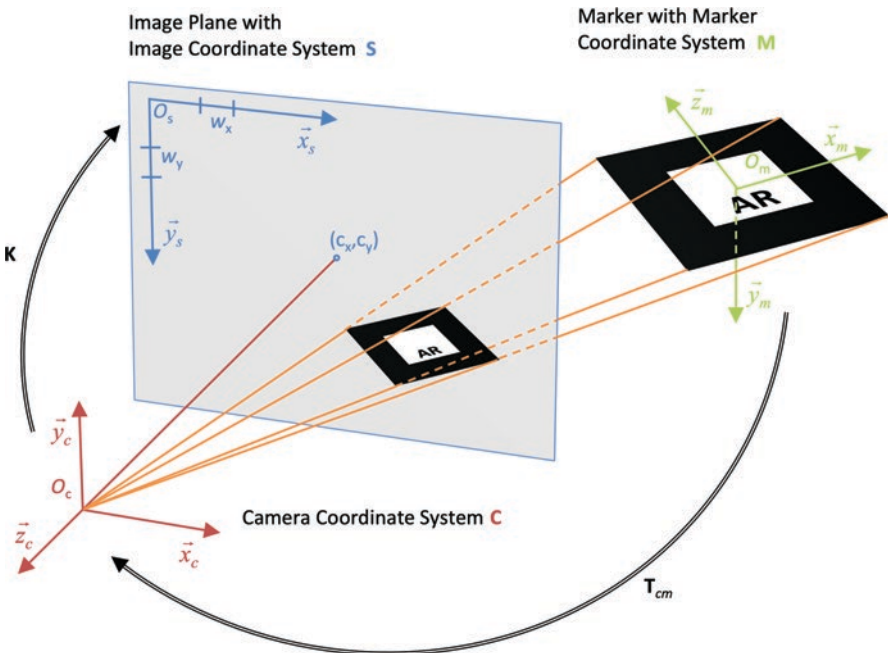

Fig. 4.11 Camera coordinate system C, image coordinate system S and marker coordinate system M (position of image plane flipped for illustration)

4.3.3   Feature-Based Tracking Techniques

In addition to marker-based tracking techniques, there are also camera-based tracking techniques that recognize features in the camera image and assign these to models. The models, which can be 2D or 3D, can be built on the fly or could be already known from a database. Feature-based tracking techniques represent a generalization of the marker-based approach.

Geometry-Based Tracking

In geometry-based tracking, features such as edges and/or vertices are extracted from the camera image. Based on an extrapolation of the transformation extracted from the previous camera image, the distances between the lines and corners of the calculated and the current image are used as the basis for the modification of the transformation.

As can easily be seen from the example of a cube with six identical sides, in many cases the individual features are not unique, i.e., there are often several valid poses for a current camera image. Thus, based on the last used pose, one of several possible transformations is always used: the one that has the smallest change to the previously calculated transformation. The correct initialization of the tracking is therefore crucial, as further poses are calculated incrementally. For a unique initialization, additional tracking techniques (such as the already described marker-based method) can be used. Neural networks are also increasingly used for matching with a given model (cf. Klomann et al. 2018).

Feature-based approaches using edges and/or corners are particularly suitable in areas of uniform geometric shapes, especially when the areas have few other features for extraction.

Other Feature-Based Tracking Techniques

Unlike corners and edges, other visual features are often not easily recognizable to a human observer. However, they offer the advantage that they can be found quickly and reliably in a camera image using corresponding feature detectors. As far as is possible to extract enough of such features from the camera image, they will be compared with existing 2D or 3D descriptions of the features (the so-called descriptor). After outliers have been sorted out – usually using a RANSAC method (Fischler and Bolles 1981) – the pose of the camera in relation to the known feature groups can be calculated on the basis of the correct assignments (see Fig. 4.12).

Feature detectors differ significantly in their speed and reliability. Not all detectors offer corresponding descriptors. It is advantageous here if the detection of the features is independent of rotation (rotation invariance) and distance (scale invariance). If this is not the case, corresponding features must be calculated from

Fig. 4.12 Assignment of feature points in the current camera image to those of an existing feature map
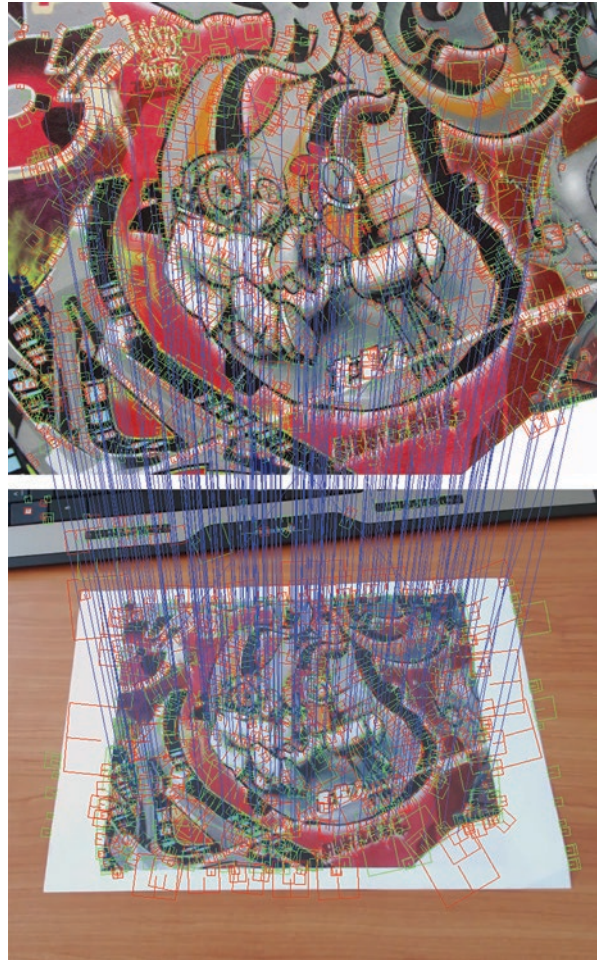
different angles and in different resolutions. Detectors used for feature-based tracking include SIFT – Scale Invariant Feature Transform (Lowe 1999, 2004) – and SURF – Speeded Up Robust Features (Bay et al. 2006). A basic description of feature-based tracking for AR can be found in (Herling and Broll 2011). Figure 4.13 shows the robustness of feature-based methods using a SURF-based approach: despite numerous occluding objects, the remaining features visible allow for a stable pose estimation.

Another possibility to implement camera-based tracking is the combined use of color cameras and depth cameras in the form of so-called RGBD cameras. Here, the depth information can be used for tracking the camera position as well as for tracking user interactions. The latter is done by estimating to what extent skeletons can fit into the recorded depth data and thus allow the recording of user movements such as the movement of an arm. RGBD cameras usually use an infrared projected

Fig. 4.13 Tracking based on features is much more robust against interference than marker-based tracking: despite numerous objects obscuring the image used for tracking, the virtual object can be registered correctly

Fig. 4.14 Projected infrared pattern for depth detection of an RGBD camera. (© DLR 2013, all rights reserved)

pattern (see Fig. 4.14) or a Time of Flight (TOF) method for depth detection, where the travel times of the reflected light are determined. The technology of RGBD cameras has become particularly well known through the great success of the first generation of Kinect, which was sold as an input device for a game console.

4.3.4   Visual SLAM

In the tracking techniques presented so far, it was assumed that markers, images, or objects are known regarding their characteristics. This made it possible to determine the relative position and orientation of the camera. If either the position and location of the markers or the camera(s) in the surrounding (spatial) coordinate system was known (e.g., in the form of a map), this information could also be used for absolute location (position estimation) in the spatial coordinate system. But how to realize a tracking in an unknown environment, i.e., without known markers, images or objects and without any information about the arrangement these in space?

In this case, SLAM (simultaneous localization and mapping) – a method originating from robotics – is used. Initially, neither the position and orientation of the camera nor the environment are known. SLAM approaches primarily based on cameras observing the environments are also referred to as Visual SLAM. For SLAM-based tracking in the AR context, either features (SIFT, SURF, FAST, etc.) and/or depth information (e.g., Kinect, Intel RealSense, Google Tango, Structure.io) are used.

More recent handheld devices may also apply LiDAR (light detection and ranging), originally used in robotics and automated driving only, providing high-quality depth estimation of the environment. While the former produce sparse maps with comparatively few feature points (cf. PTAM, Klein and Murray 2007), the latter generally use dense maps of volume. Since initially no map exists, the coordinate system can be freely selected based on the starting position. The map is then successively created based on the movement of the camera, i.e., features found in the current camera image are compared with the existing map and new features are located in the map. Based on the already known parts of the map, the position and location of the camera are simultaneously reassessed based on detected features.

The simultaneous reconstruction of the environment in the form of a map as well as the estimation of the position based on this still incomplete information usually leads to increasing errors (both with regard to the quality of the map and the position estimation based on it) as long as new unknown areas are continuously added. It is crucial that known surrounding areas are reliably recognized, even if their position and location are different from the current map information. In this so-called loop closing, all map data must be adjusted to ensure that the current and stored information are consistent.

Dynamic objects represent an additional difficulty with SLAM methods. Since the resulting features change their position and location, they must be identified and then ignored in the processing, otherwise they lead to both a faulty map and faulty tracking.

4.3.5   Hybrid Tracking Techniques

For augmented reality applications it is common to use combinations of different tracking techniques. The reason for this is usually that the individual methods provide different results, depending on the situation. A typical example is a marker-based approach: this approach usually works well if the position and location in relation to the camera can be determined for all virtual content via at least one marker. However, if an occlusion occurs even for a short time, the marker is not recognized and registration of the virtual object(s) in the real scene is no longer possible. In order not to immediately lose the illusion of an augmented reality, it is therefore recommended to estimate the change of position and attitude based on alternative tracking techniques. If, for example, a tablet or smartphone is used, the change in position could also be determined by the integrated position sensors (see Sect. 4.2.3). This can be used to ensure that in situations where the brand tracking does not provide any information, a transformation can be specified that is correct at least regarding the position. If the user does not change his or her position until the corresponding marker is visible again, or only changes it slightly, the illusion can be maintained in this way.

Another way to compensate for short-term failures or even latency of the tracking technique is to use prediction techniques. While simple extrapolation methods are basically also suitable for this purpose, Kalman filters (cf. Bishop et al. 2001, p. 81) are a widely used and significantly better alternative. Depending on whether the position or the rotation must be estimated, ordinary or advanced Kalman filters are used. Another possibility is the use of particle filters (cf. Arulampalam et al. 2002).

Cloud-Based Tracking

Hybrid tracking techniques can also be used for multiuser experiences. The first step is to build a tracking reference (called an anchor) within a spatial environment or context. Feature maps in combination with additional information like GPS data (for outdoor applications) or room information (for indoor applications) can be used for this. The second step is to send to a cloud service. By downloading this cloud anchor, applications on other devices can align virtual objects to the same spatial context, enabling users to view the same content at the same location but from an individual perspective (see Fig. 4.15, left).

In visions of the near future of computing – coined as AR Cloud, Spatial Web, Mirror World or Digital Twin – a large amount of constantly updated digital content (e.g., construction, IoT, traffic, shops, artists) is spatially anchored and can be perceived and shared by many users as a persistent, dynamic overlay of the real world (see Fig. 4.15, right). Reliable, precise and easily functioning tracking and localization technologies are an essential part of the implementation of these concepts. Organizations are developing universal open standards to ensure open, free and
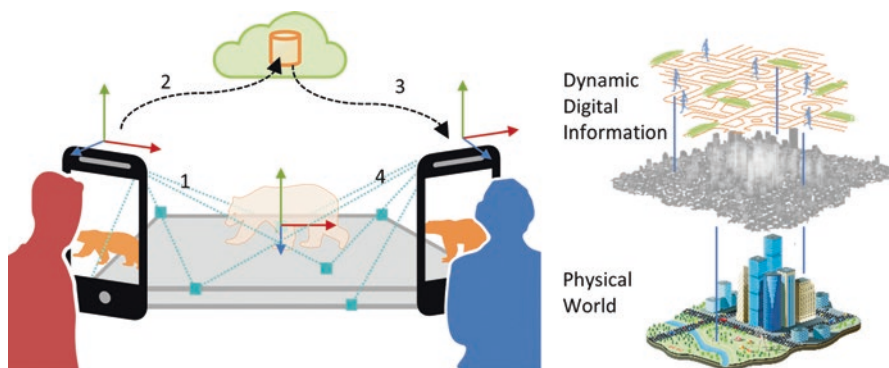
Fig. 4.15 (Left) Simplified concept of cloud anchors: 1) One device captures peculiar features from the environment. 2) It saves these and an object anchor in cloud storage. 3) Another device downloads this information and 4) tries to find the same features in its view to position a virtual object at the same anchor position. (Right) The AR Cloud concept: different layers of dynamic georeferenced information augment the real world

interoperable use of the deeply linked partial technologies. For example, the Open AR Cloud organization (OpenAR 2021) together with the Open Geospatial Consortium (OGC), is developing a standard for a geographically anchored poses with six degrees of freedom (GeoPose 2021) referenced to standardized Coordinate Reference Systems (CRSs). Since these tracking and immersive visualization technologies capture and operate with many personal and potentially protected private data, for long-term acceptance it is important to take care of privacy and data security issues and to respect possible ethical, legal and social impacts (CyberXR 2021) as part of development and operation.

Microsoft Hololens Tracking

The SLAM approach used in Microsoft's Hololens 2 (see Hololens 1 in Fig. 5.10) has several special features regarding the combination of different hardware sensors. It uses a total of four cameras exclusively for tracking. The four cameras work with a comparatively low frame rate of only 30 Hz. This means that fast head movements cannot be detected without noticeable latency. To compensate for fast movements, the tracking data is therefore combined with those of an IMU (see Sect. 4.2.3) with an update rate of 1000 Hz. This allows not only the calculation of intermediate values between the determined camera poses at 240 Hz, but also compensation of color shifts (late-stage reprojection) due to the color sequential output (see Sect. 5.3.2). Instead of a global coordinate system, a graph of position estimations is used, whereby the individual local coordinate systems are connected by relative poses. If relative poses are not, or not yet, available, the graph may break up into several subgraphs. A loop closing does not take place, so that the graph is not necessarily globally consistent.

In addition, data from a depth camera (1-MP Time-of-Flight depth sensor) is used for spatial mapping with a framerate of 1–5 fps. If a user's hand is recognized the modus of the depth camera will change to high-frame rate (45 fps) near depth sensing, which is used for hand tracking in an area up to 1 m (see also Sect. 4.4). For power saving, it reduces the number of illuminations while doing the hand tracking.

Furthermore, the Hololens has a high-resolution front camera with a FOV of 65°, a five-channel microphone array with noise cancellation to allow voice input even in loud environments, and eye tracking (see Sect. 4.5). The eye tracking is especially used for the rendering using the waveguide displays (see Sect. 5.3.2).

4.4 Finger Tracking

Although the interaction with standard input devices and the corresponding interaction methods are usually sufficient, these devices and methods hardly reproduce the natural interaction of a human being with the virtual world. New types of interaction (e.g., by pointing gestures) must first be explained to the user.

One example is the virtual assembly simulation. Using a standard interaction device such as a VR controller, a component can be easily moved from one location to another by detecting its position and orientation and by pressing a button. However, it is not possible (or very difficult) to check whether a user is able to install a component with only one hand or whether the user needs both hands for this action. Figure 4.16 left shows a user in front of a VR display during a virtual assembly simulation of a satellite. The user is equipped with optically tracked 3D glasses and a finger tracking device and tries to insert a module of the satellite with only one hand into the corresponding module slot. Other scenarios in the field of virtual assembly simulation are testing for the general tangibility of objects or the
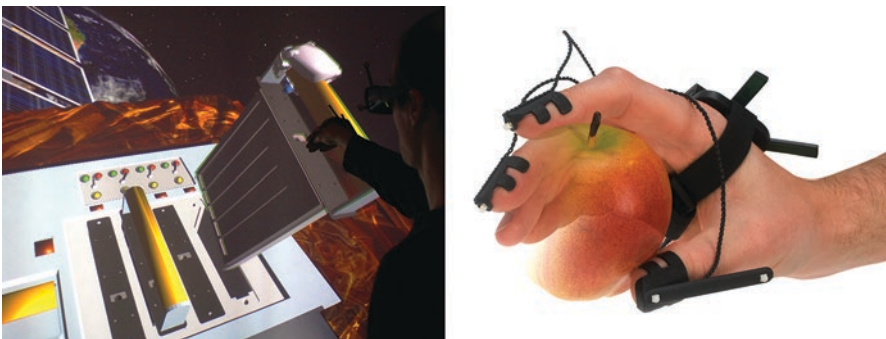

Fig. 4.16 (Left) User with tracked 3D glasses and finger tracking during an installation test of a satellite module in a virtual assembly simulation (© DLR 2013, all rights reserved). (Right) Grasping a virtual apple with a tracked hand (© ART 2013, all rights reserved)

transfer of objects from one hand to the other. The use of standard interaction devices like VR Controller is not suitable for this kind of applications.

In general, the direct interaction of users with their environment by tracking their hands and fingers in the virtual world is easier and more intuitive for them (Bowman et al. 2004). In contrast, interactions with VR are faster when using indirect interaction methods in combination with simple or standard interaction devices (Möhring and Fröhlich 2011; Hummel et al. 2012).

In general, the term finger tracking is used to describe the detection of the position and usually also the orientation of a hand and its fingers. Depending on the application, the required accuracy varies. Relatively low accuracy and only the detection of the position of the back of the hand or a finger is already sufficient to emulate a mouse or to interact with a user interface in a virtual world. However, low to medium accuracy and the relative position of individual fingers to each other is already necessary to recognize gestures. For application areas such as virtual assembly simulation in the automotive, aerospace and aviation industries, which require direct interaction, not only the position and orientation of the back of the hand and all fingertips are important for tracking, but also the lengths of the individual finger links and the angles of the corresponding finger joints. Only this accuracy enables a perfect image of the real hand.

There are two major challenges in finger tracking. First, the human hand has many degrees of freedom. The back of the hand is usually seen as a rigid body with six DOF: three translational and three rotational (see Fig. 4.17). Each finger has another four DOFs, two rotational DOFs at the root of the finger and one rotational DOF each for the joints to the middle and outer phalanx. The thumb has a special role because it has an additional DOF at the root. Therefore, five DOFs are required for the thumb, three rotary DOFs at the wrist and one for each additional finger joint. Added up, this results in 27 DOF for one hand (Lin et al. 2000). Second, the tight position of the fingers in relation to each other is a great challenge for the tracking system. For optical systems in particular this is a non-trivial problem to
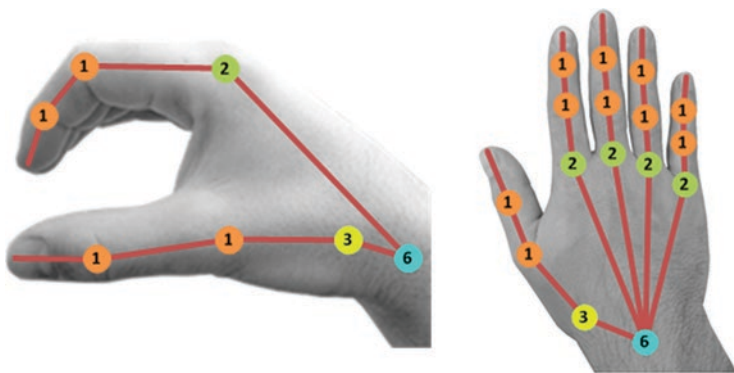

Fig. 4.17 Data model of a hand to implement finger tracking (the circles symbolize the joints of the hand and fingers with their respective degrees of freedom; the lines represent the skeleton)

solve because of the occlusion of markers, the small visual difference of the fingers and the 27 DOF per hand.

In addition, it should not be forgotten that each person's hands and fingers are different. This includes not only the length and thickness of the individual phalanges, but also the joints and joint angles between them. A physical handicap or even the absence of one or more fingers must not be ignored either. The respective tracking devices must take this into account and be adaptable to it.

Since finger tracking has high requirements on the tracking hardware, a wide variety of techniques are employed. In earlier days mechanical tracking techniques were most common, for example optical fibers, strain gauges or potentiometers (variable resistors). The Sayre Glove (DeFanti and Sandin 1977) has bendable tubes that run along each finger inside a glove. The Data Glove (Zimmermann et al. 1986) uses two optical fibers per finger. At one end of this fiber optic cable is a light source; at the other end is a photocell. Depending on the bending of the finger, a different amount of light hits the photocell. This allows the joint angles of the fingers to be approximately determined. The CyberGlove (Kramer and Leifer 1989) uses 22 thin, metallic strain gauges to measure the joint angles of the fingers. In the Dexterous Hand Master (Bouzit et al. 1993), an exoskeleton is pulled over the hand and fingers. Using cable pulls, potentiometers are then activated, from whose resistance values the positions of the fingers can be determined by analog/digital converters. With mechanical methods, however, only a relative measurement of the fingers to the back of the hand is possible. The position and orientation of the back of the hand must be measured using a different tracking technique.

More rarely, magnetic trackers are used for finger tracking. These can detect up to 16 individual 6-DOF sensors. This means there is one sensor for each of the three-finger links and one sensor for the back of the hand. The disadvantage of magnetic tracking is the slight susceptibility to interference from metallic or electromagnetic sources. In addition, most magnetic trackers are wired due to their design.

Optical finger-tracking devices predominate in the non-mechanical tracking techniques. The MIT LED Glove (Ginsberg and Maxwell 1983) is equipped with light-emitting diodes (LEDs), which are recorded by an external camera system. To distinguish individual fingers from each other, the LEDs flash alternately one after the other (Hillebrand et al. 2006). At a recording rate of 60 Hz, for example, the alternate flashing of the LEDs reduces the repetition frequency to 20 Hz for a three-finger system and to 12 Hz for a five-finger system. The use of optical tracking enables high accuracy and lightweight wireless interaction devices, but usually at least four expensive special cameras are required to ensure triangulation of each LED used. Some optical finger tracking devices are additionally equipped with inertial sensors to temporarily bridge any obscurations of the LEDs, which often occur due to the small distances between the fingers. In Hackenberg et al. (2011) a method was presented that is based on depth cameras and uses special feature detectors for finger phalanges and fingertips.

There are inexpensive camera-based finger trackers available, which nevertheless offer high accuracy and low latency and can be easily integrated into VR applications. Leap Motion, as an example, uses two cameras in combination with infrared

Fig. 4.18 3D model of a hand controlled by a VR controller with touch sensors

LEDs (wavelength 850 nm). The hardware covers an interaction space of up to 80 cm by 80 cm, with the brightness of the infrared LEDs being the limiting factor. The controller transmits two grayscale videos to the software, which in turn determines the finger positions from this data. Usually, the controller is used while lying on a table. With the help of an adapter, however, it is also possible to attach the controller to VR glasses to use finger gestures as input for VR applications.

Using touch-sensitive surfaces it is also possible to track fingers using a VR controller (see Fig. 4.18).

4.5 Eye Tracking

4.5.1 Eye Movements

Eye-tracking, or gaze registration, generally refers to tracking the movement of the human eye. The procedure is used to record and evaluate the course of a person's gaze.

If a user views an image, he focuses by changing the focal length of his lens and depicts the image onto light-sensitive cells of the retina. The amount of incident light can be varied through the iris. The iris works like an aperture and changes the diameter of the pupil. The eye muscles that move the eye in the eye socket are attached to the sclera. The types of movement of the eye are differentiated into drifting, following, trembling, rotating, fixing and saccades. However, only the last two are interesting for tracking the eye. During fixation, e.g., while reading, the eye concentrates on one point and collects information. Saccades are jumps that take place between fixation and last about 20 ms to 40 ms.

4.5.2 Methods

Various technical methods have been developed in recent decades to determine the direction of gazes. An overview of these methods and sub methods is given in Fig. 4.19. In principle, a distinction is made between invasive and non-invasive
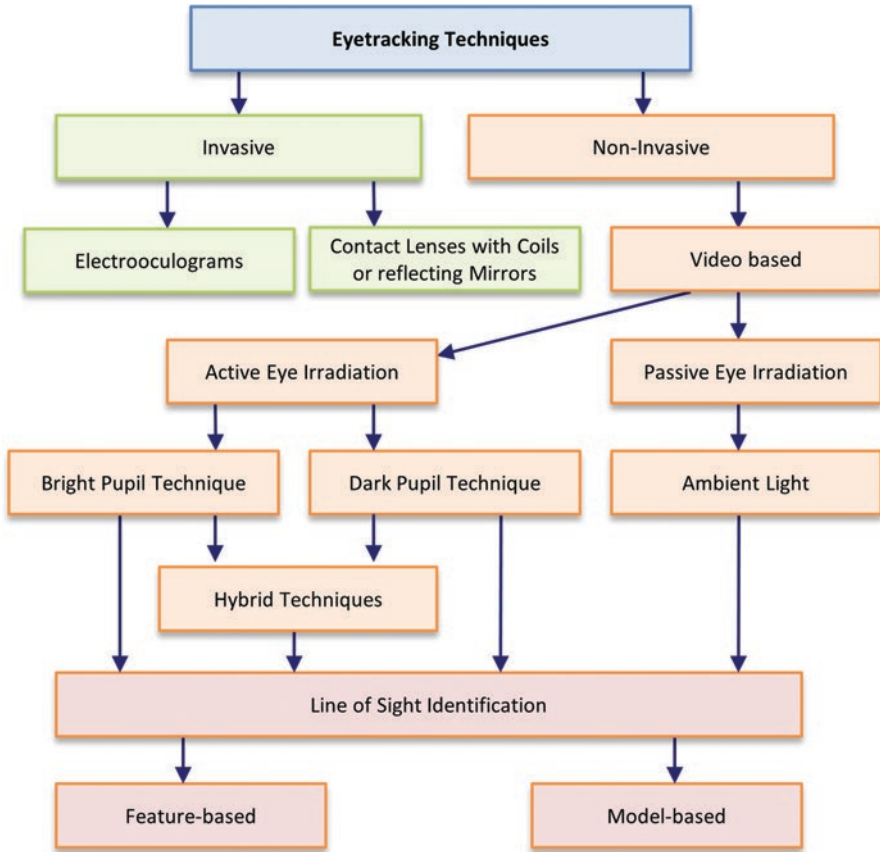
Fig. 4.19 Overview of methods for eye tracking

procedures. Invasive procedures always require a direct intervention on the user's body, e.g., with electrodes.

With non-invasive procedures the user's gaze can be followed without contact. The first developed eye-tracking techniques were purely invasive. Electrooculography was developed more than 40 years ago. In electrooculography, the electrical potentials of the skin around the eye are measured. These potentials range from 15 µV to 200 µV. The sensitivity for eye-tracking is about 20 µV/angle degree (Duchowski 2007). With this technique the relative eye movement to the head can be recorded. However, it is not possible to determine an absolute point of view of the eye on an object. Another invasive eye-tracking technique is the contact lens method. Here, contact lenses are used either with small coils or with reflectors. For contact lenses with coils, the change of the magnetic field is measured, and from this the relative movement of the eye is derived. If there are reflectors on the contact lenses, the reflected light can be used to deduce the relative direction of vision.

In recent years non-invasive video-based eye-tracking techniques have been used. Here, the eye is captured by a camera and the gaze direction is determined by image processing algorithms. In video-based methods, a distinction is made between passive and active eye irradiation. Passive methods use ambient light to irradiate the eye scene. Due to the undefined irradiation conditions of an environment, there are high requirements for precise feature identification of the eye components.

With passive irradiation, the contour between the dermis and iris is used to identify features. A more precise method is the active irradiation of the eye scene by an infrared light source. Figure 4.20 illustrates the more favorable contrast ratios of the active method, which enables robust feature identification between pupil and iris.

Depending on the arrangement of the IR irradiation source, a distinction is made in active irradiation procedures between the light and dark pupil technique. If the irradiation source is located outside the optical axis of the eye-tracking camera, the radiation is reflected by the iris and sclera; thus, the pupil is the darkest object within the recorded eye scene. If the light source and the camera are arranged in the same optical axis, the radiation is reflected at the retina inside the eye, making the pupil the brightest object.

Hybrid processes require optics with different arrangements of the IR irradiation sources. Regardless of whether active or passive eye irradiation is used, the evaluation of the direction of gaze is based on features on the one hand and on models on the other. Combined methods are also used. Feature-based methods detect contours, e.g., the pupil geometry, and calculate the center point and the relative gaze coordinates. Side effects, such as reflections, can cause other features to be interpreted as the pupil; this property reduces the accuracy of feature-based methods. Model-based methods, on the other hand, compare the image information of the recorded eye scene with a corresponding model of an eye. By varying the parameters, an iterative attempt is made to adapt the model to the real eye scene. If the model could be adapted with a certain error, the relative gaze coordinates are obtained. Model-based procedures belong to the more precise, but also to the more computationally intensive, approaches. Video-based eye-tracking techniques not
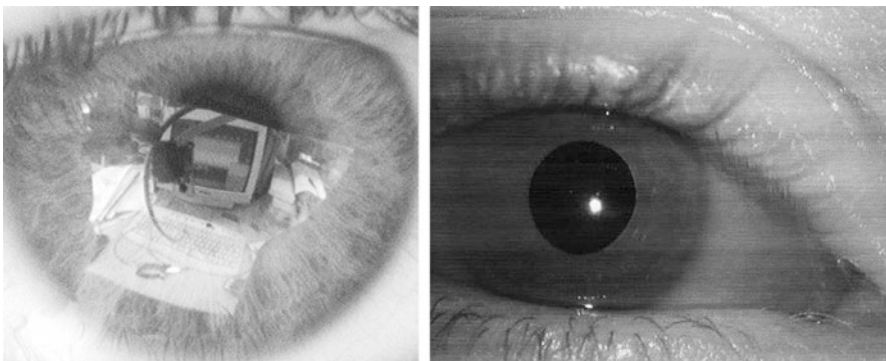

Fig. 4.20 Recorded eye scene with passive and active irradiation

only allow the relative direction of gaze to be determined. With calibration, the correspondence between the direction of gaze and regions in the virtual image (e.g., a button) can be found.

4.5.3   Functionality of an Eye Tracker

Figure 4.21 shows the basic procedure of an eye-tracking routine with active illumination and bright pupil technique. An eye-tracking camera, which is focused on the user's eye, captures a digital grayscale image. This image is passed to the eye-tracking image processing system. First, an adjustment of the gray values is applied and then the image is pre-filtered, e.g., to improve a noisy image. Furthermore, a
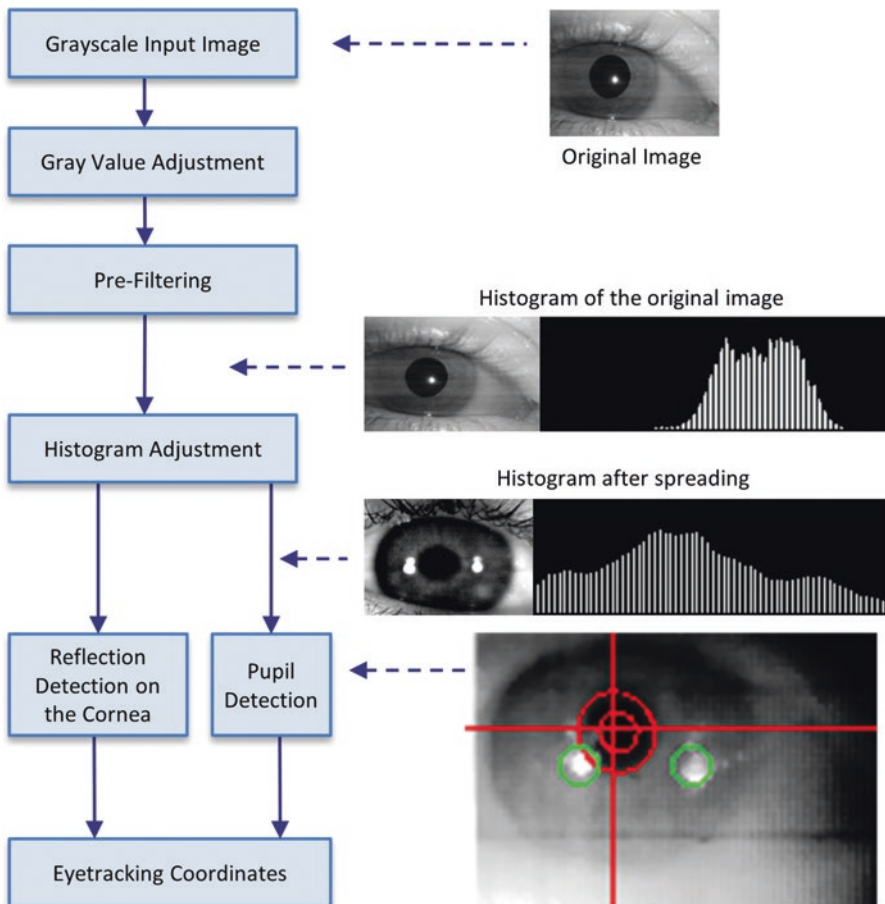

Fig. 4.21 Image processing process for eye tracking

histogram spread is performed to highlight the object contours of the eye such as the pupil or iris. In the next step, the contour of the pupil is detected by edge detection, and the pupil center is calculated. Furthermore, in the case of active illumination, the reflections at the cornea are used as additional information. With a Head-Mounted Display (HMD; see Chap. 5) with integrated eye tracking, these reflections are often used as a reference point. The eye-tracking image processing finally outputs the coordinates of the pupil center in horizontal and vertical direction. If the corneal reflections are also evaluated, the eye-tracking image processing outputs a difference vector between the pupil center and the center of the corneal reflection, from which it can be concluded where the user focuses.

4.5.4   Calibration

To enable user interaction with virtual objects in addition to the actual eye-tracking, an assignment between the camera's detection range and the displayed image is necessary.

Figure 4.22 shows the nested coordinate systems of the eye-tracking camera and the virtual image. To be able to establish a connection between the pair of coordinates in the camera coordinate system $\vec{x}_c$, $\vec{v}_c$ and the coordinates of the virtual image $\vec{x}_{virt}$, $\vec{v}_{virt}$, there are various assignment methods. In Duchowski (2007) a simple linear analytical mapping function is presented. Equations (4.1) and (4.2) describe the linear mapping functions for the horizontal and vertical direction. In eq. (4.1) the horizontal coordinate $\vec{x}_c$ is set by subtracting from $\vec{x}_{c\_min}$ to its origin. Then this coordinate is scaled to the virtual image by the horizontal resolution ratio between the virtual coordinate system and the camera coordinate system. Then the relative position in the virtual image is calculated by adding the minimum coordinate of the virtual image. For the vertical coordinate assignment, the calculation method described in eq. (4.2) is analogous to eq. (4.1).

$$\vec{x}_{virt} = \vec{x}_{virt\_min} + \frac{\left(\vec{x}_c - \vec{x}_{c\_min}\right)\left(\vec{x}_{virt\_max} - \vec{x}_{virt\_min}\right)}{\left(\vec{x}_{c\_max} - \vec{x}_{c\_min}\right)} \tag{4.1}$$

$$\vec{y}_{virt} = \vec{y}_{virt\_min} + \frac{\left(\vec{y}_k - \vec{y}_{c\_min}\right)\left(\vec{y}_{virt\_max} - \vec{y}_{virt\_min}\right)}{\left(\vec{y}_{c\_max} - \vec{y}_{c\_min}\right)} \tag{4.2}$$

In practice, more complex assignment procedures, such as the second and third-order polynomial procedure or the homographic procedure, are usually used. These assignment procedures require several parameters. The parameters are obtained by a calibration routine. In this calibration routine the user has to select points that are distributed over the virtual image (e.g., in the corners and in the middle). The user must look at these points one after the other. Using these parameters, the calibration routine can now determine the parameters for the complex assignment functions.
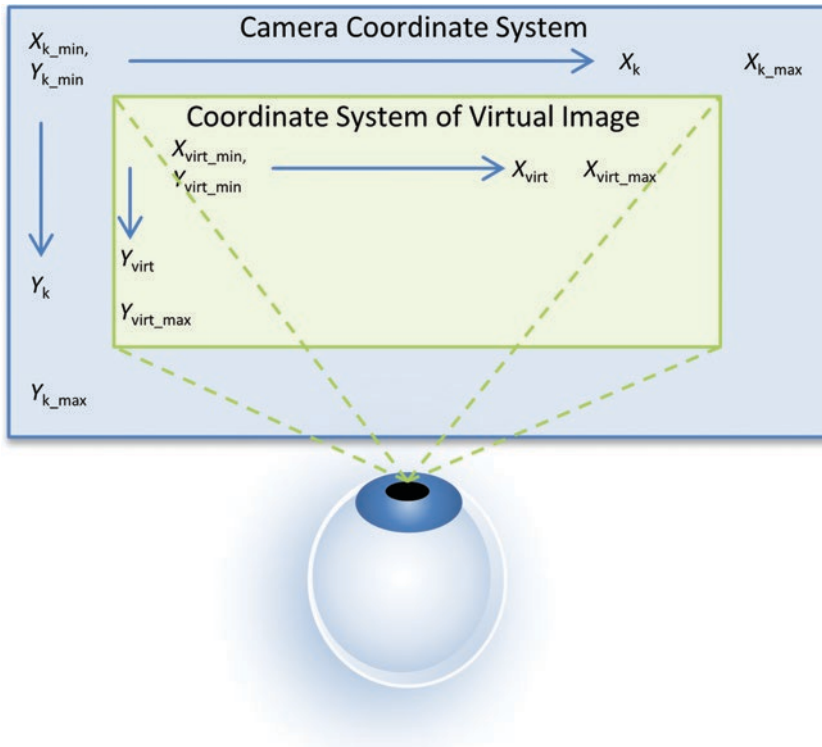
Fig. 4.22 Camera coordinate system of the virtual image and the camera

4.5.5   Eye Tracking in Head-Mounted Displays

If you want to use gaze control, you can use an eye-tracking HMD. Figure 4.23 shows the basic procedure of an eye-tracking HMD. As already mentioned in Sect. 4.5.2, a camera is required for a video-based procedure. The camera is attached to the HMD in a way that it can focus on the eye. The captured image of the eye scene is then transmitted to the computer or to the HMD electronics and an eye-tracking algorithm calculates the direction of the eye (see Sect. 4.5.3).

Eye-tracking HMDs evaluate either both eyes simultaneously or only one eye. If, for example, the gaze direction of both eyes is determined, the 3D viewpoint of the user can be determined from the intersection of both vectors.

As already explained in Sect. 4.5.4, there must be a correspondence between the coverage area of the camera and the display area of the virtual projection. Therefore, a calibration must be carried out. Compared to the remote eye trackers presented in Sect. 4.5.6, eye tracking HMDs have better conditions for recalibration due to the tight fit of the glasses. If the HMD moves only slightly, the calibration does not have to be repeated during operation.
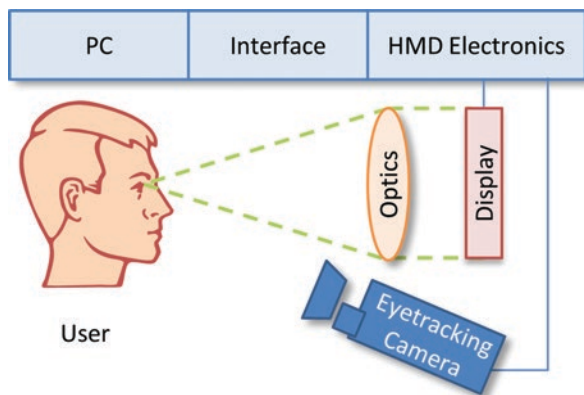
Fig. 4.23 Basic procedure of an eye-tracking HMD

4.5.6 Remote Eye Tracker

A remote eye tracker has essentially the same components as the eye-tracking HMD presented in Sect. 4.5.5. With a remote eye tracker, the user sits in front of a monitor. A camera mounted near the monitor focuses on the user's head. There are two methods to capture one or both eyes. On the one hand the camera captures a large area where the user's head is located. The image processing locates the area of the eye and calculates the position of the pupil in this section. With this method, only a few pixels are available to calculate the pupil position. This low resolution of the pupil area also reduces the accuracy. With a second method, the eye-tracking camera captures only a small area, but this area is captured with high resolution. This camera automatically aligns itself so that the current position of the eye is recorded. As mentioned in Sect. 4.5.4 a calibration must be performed for the remote eye tracker to assign the calculated coordinates of the gaze direction to the display area of the monitor. Unlike eye-tracking HMDs, remote eye trackers often need to be recalibrated during operation because the user changes their sitting position relative to the monitor and the eye-tracking camera.

4.6 Further Input Devices

In this section we will consider other input devices that are often used to build VR systems, in addition to standard PC input devices (such as 2D mouse, keyboard, microphone or touch monitors).

Fig. 4.24 Different variants of a 3D mouse

4.6.1   3D Mouse

One of the simplest input devices is the 3D mouse (see Fig. 4.24). This enables direct navigation according to the six degrees of freedom as well as interaction via freely assignable buttons. By shifting the mouse sideways and pushing and pulling it vertically, a translation in 3D space can be performed; by turning or tilting it, a corresponding rotation is achieved.

Versions of the 3D mouse differ not only in size but also in the integration of additional buttons, which are usually freely assignable. The advantage of a 3D mouse is its high accuracy. Because a 3D mouse is usually placed on a table, it is more suitable for desktop VR. Sometimes it is also used as a control unit permanently mounted on a column, which limits the user's working range.

4.6.2   Mechanical Input Devices

Mechanical input devices record the movements of a user via a mechanism (e.g., via a linkage or cable pulls). The advantage of mechanical input devices is that, on the one hand that they can be highly accurate, and on the other hand that they are well suited to provide haptic feedback to the user. The disadvantages are that the user always has something in his or her hand or has to be connected in some way to the mechanical input device and that the mechanics may be a disruptive object. Figure 4.25 shows an example of a mechanical input device where the user holds a pen in his or her hand. The fact that the user is used to holding pens means that the use of the device can become part of normal habits, provided that the actual application supports this usage scenario.

Mechanical input devices use angle or distance measurements at the joints to obtain users interactions. The high accuracy is achieved by correspondingly accurate angle measurements, which are usually carried out using gear wheels or gears, potentiometers, or strain gauges. In some cases, similar measuring methods are used as in computer mice, which are known to allow high resolution. The latency of mechanical input devices is low due to the direct measurement. Smooth operation is particularly important for use (Salisbury and Srinivasan 1997) in order not to be

Fig. 4.25 Mechanical input device in pen form with haptic feedback

restricted by the input device and thus to perceive it as disturbing. By integrating haptic feedback, a mechanical input device becomes an output device at the same time (see end effector displays in Sect. 5.5).

4.6.3 Treadmills for Virtual Reality

Due to the limited size of a VR system, it is difficult to allow walking or running around in a virtual environment. In most cases the user reaches the edge of the interaction area after a few steps. Accordingly, control techniques for navigation have established themselves, using different input devices such as VR controllers or a flystick (see Sect. 4.3.1). In addition, input devices have been developed that allow walking or a walk-like movement for navigation in virtual worlds. Many approaches are based on the idea of treadmills on which users move and whose speed is controlled by the VR system. By means of a mechanism for tilting, it is possible to walk uphill or downhill. The disadvantage of treadmills, which are used in a similar way in gyms, is that they only allow walking or running in one direction, which is a significant limitation for use in VR systems.

In recent years, so-called omnidirectional treadmills have been developed using different approaches. One possibility is to construct the treadmill from small treadmills that are arranged orthogonally to the main direction. This creates a surface on which the user can move in all directions. By tracking the user, the individual treadmills can be controlled so that the user always moves in the middle of the surface. The CyberWalk Treadmill (Souman et al. 2008) is an

example of this. Large balls, in which the user moves and which are themselves supported so that they remain in one place, are another possibility. The problem with this approach is that the perceived floor for the user is not flat but curved by the shape of the sphere. This can make walking more difficult. The Cybersphere (Fernandes et al. 2003) is an example of this type. Other variants are based on constructing the floor from appropriately arranged castors to allow walking around. More cost-effective approaches are based on the idea of holding the user in place by means of a retaining ring and allowing him or her to walk on a smooth or slippery floor. The Virtuix Omni (see Fig. 4.26) and the Cyberith Virtualizer are examples of this.

Fig. 4.26 User with VR glasses on an omnidirectional treadmill

4.7   Summary and Questions

In this chapter you have acquired basic knowledge in the field of tracking and VR/ AR input devices. Starting from the consideration of how many degrees of freedom an object has, basic terms such as accuracy, repetition rates, latency and calibration were introduced with respect to their applicability in the fields of VR and AR. Following the presentation of different tracking techniques for the continuous determination of 3D data, further input devices were introduced.

Check your understanding of the chapter by answering the following questions:

- Why is high accuracy not sufficient as a requirement for VR/AR input devices?
- Which effects can cause problems during data acquisition?
- What is determined by a tracking system and what are the characteristics of tracking systems?
- What effects can interfere with a tracking system?
- What problems arise with outdoor tracking in city centers and what alternatives exist?
- Find an application example for hybrid tracking techniques.
- What is the difference between inside-out and outside-in tracking techniques and what are their advantages and disadvantages?
- What are the advantages of camera-based tracking?
- Why should you actively illuminate the eyes of a user during eye tracking and what should be considered?
- How many degrees of freedom must be determined for finger tracking?

Recommended Readings

Bishop G, Allen D, Welch G (2001) Tracking: beyond 15 minutes of thought. SIGGRAPH 2001, Course 11, http://www.cs.unc.edu/~tracker/media/pdf/ SIGGRAPH2001_CoursePack_11.pdf. Accessed 18 October 2021 – *The authors of this course at SIGGRAPH presented a good overview of the technical fundamentals of input devices.*

Szeliski R (2011) Computer vision - algorithms and applications, *Springer, DOI* https://doi.org/10.1007/978-1-84,882-935-0. – *This book gives a good overview of the basics of computer vision, which is used for VR and AR.*

References

Abawi FA, Bienwald J, Dörner R (2004) Accuracy in optical tracking with fiducial markers: an accuracy function for ARToolKit. In: Proceedings of the 3rd IEEE/ACM international symposium on mixed and augmented reality (ISMAR '04). IEEE Computer Society, Washington, DC, USA, pp 260–261. https://doi.org/10.1109/ISMAR.2004.8

Arulampalam MS, Maskell S, Gordon N, Clapp T (2002) A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans Signal Process 50(2):174–188

Bay H, Tuytelaars T, Van Gool L (2006) SURF: speeded up robust features. In: Computer vision–ECCV 2006. Springer, Berlin Heidelberg, pp 404–417

Berry R, Billinghurst M, Cheok AD, Geiger C, Grimm P, Haller M, Kato H, Leyman R, Paelke V, Reimann C, Schmalstieg D, Thomas B (2002) The First IEEE international augmented reality toolkit workshop. IEEE Catalog Number 02EX632

Bishop G, Allen D, Welch G (2001) Tracking: Beyond 15 minutes of thought, SIGGRAPH 2001, Course 11. http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_11.pdf. Accessed 18 March 2021

Bouzit M, Coiffet P, Burdea G (1993) The LRP Dextrous Hand Master. Proceedings of Virtual Reality Systems Fall '93, New York

Bowman DA, Kruijff E, LaViola JJ, Poupyrev I (2004) 3D-user interfaces: theory and practice. Addison Wesley Longman Publishing Co., Inc., Redwood City

CyberXR (2021) Cyber-XR Coalition: Immersive technology standards for accessibility, inclusion, ethics and safety. https://www.cyberxr.org, Accessed 18 Mar 2021

DeFanti IA, Sandin DJ (1977) Final report to the National Endowment of the Arts. US NEA R60–34-163, University of Illinois at Chicago Circle, Chicago, IL

Duchowski A (2007) Eye tracking methodology: theory and practice. Springer, London

Fernandes KJ, Raja V, Eyre J (2003) Cybersphere: The fully immersive spherical projection system. Communications of the ACM, 46(9), 141–146. ACM, New York

Fischler MA, Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381–395. ACM, New York

GeoPose (2021) GeoPose Standards Working Group, https://www.ogc.org, Accessed 18 Mar 2021

Ginsberg CM, Maxwell D (1983) Graphical marionette. Proceedings of SIGGRAPH Computer Graphics 18(1):26–27

Goldstein H (1980) Classical mechanics. Addison-Wesley

Hackenberg G, McCall R, Broll W (2011) Lightweight palm and finger tracking for real-time 3D gesture control. In Proceedings of IEEE Virtual Reality Symposium 2011 (IEEE VR 2011), pp. 19–26

Hartley R, Zisserman A (2000) Multiple view geometry in computer vision. Cambridge University Press, Cambridge

Herling J, Broll W (2011) Markerless tracking for augmented reality. In: Handbook of augmented reality. Springer, New York, pp 255–272

Herout A, Zacharias M, Dubská M, Havel J (2012) Fractal marker fields: No more scale limitations for fiduciary markers. In IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 285–286. IEEE

Hillebrand G, Bauer M, Achatz K, Klinker G (2006) Inverse kinematic infrared optical finger tracking. 9th International Conference on Humans and Computers (HC 2006). Key 1045432

Hummel J, Wolff R, Dodiya J, Gerndt A, Kuhlen T (2012) Towards interacting with force-sensitive thin deformable virtual objects. Joint Virtual Reality Conference of ICAT – EGVE – EuroVR, 2012 Eurographics Association, pp. 17–20. https://doi.org/10.2312/EGVE/ JVRC12/017–020

Kato H, Billinghurst M (1999) Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR), pp. 85–94. IEEE

Klein G, Murray D (2007) Parallel tracking and mapping for small AR workspaces. In 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 225–234, IEEE

Klomann M, Englert M, Weber K, Grimm P, Jung Y (2018) Improving mobile MR applications using a cloud-based image segmentation approach with synthetic training data. In Proceedings of the 23rd International Conference on 3D Web Technology, Web3D 2018, pp. 4:1–4:7. ACM

Köhler J, Pagani A, Stricker D (2010) Detection and identification techniques for markers used in computer vision visualization of large and unstructured data sets. In Applications in Geospatial Planning, Modeling and Engineering (IRTG 1131 Workshop)

Kramer J, Leifer L (1989) The talking glove: An expressive and receptive 'verbal' communication aid for the deaf, deaf-blind, and non-vocal. Proceedings of the 3rd Annual Conference on Computer Technology, Special Education, Rehabilitation. California State University Press, Northridge

Lee JC (2008) Hacking the Nintendo Wii remote. Pervasive Computing 7(3):39–45. https://doi.org/10.1109/MPRV.2008.53

Lin J, Wu Y, Huang TS (2000) Modeling the constraints of human hand motion. In Proceedings of the Workshop on Human Motion (HUMO '00), IEEE Computer Society, Washington, DC, USA

Lowe DG (1999) Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Vol. 2, pp. 1150–1157. IEEE

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110

Möhring M, Fröhlich B (2011) Effective manipulation of virtual objects within arm's reach. 2011 IEEE Virtual Reality Conference, 2011, pp. 131–138. IEEE. https://doi.org/10.1109/VR.2011.5759451

OpenAR (2021) Open AR Cloud. https://www.openarcloud.org/. Accessed 18 March 2021

Salisbury JK, Srinivasan MA (1997) Phantom-based haptic interaction with virtual objects. Computer Graphics and Applications. IEEE. https://doi.org/10.1109/MCG.1997.1626171

Schmalstieg D, Höllerer T (2016) Augmented reality: Principles and practice. Pearson

Souman JL, Robuffo Giordano P, Schwaiger M, Frissen I, Thümmel T, Ulbrich H, Bülthoff HH, Erst MO (2008) Cyberwalk: Enabling unconstrained omnidirectional walking through virtual environments. ACM Transactions on Applied Perception. https://doi.org/10.1145/2043603.2043607

Szeliski R (2011) Computer vision - algorithms and applications, Springer. https://doi.org/10.1007/978-1-84,882-935-0

Zimmerman TG, Lanier J, Blanchard C, Bryson S, Harvill Y (1986) A hand gesture interface device. Proceedings of SIGCHI Bulletin, 17, SI(May 1987), 189–192