Chapter

Mixed Reality: A Known Unknown

Branislav Sobota, Štefan Korečko, Marián Hudák and Martin Sivý

Abstract

Mixed reality (MR) is an area of computer research dealing with the combination of real-world and computer-generated data (virtual reality), where computer-generated graphical objects are visually mixed into the real environment and vice versa in real time. This chapter contains an introduction to this modern technology. Mixed reality combines real and virtual and is interactive, real-time processed, and registered in three dimensions. We can create mixed reality by using at least one of the following technologies: augmented reality and augmented virtuality. The mixed reality system can be considered as the ultimate immersive system. MR systems are usually constructed as optical see-through systems (usually by using transparent displays) or video see-through. Implementation of MR systems is as marker systems (real scene will be added with special markers. These will be recognized during runtime and replaced with virtual objects) or (semi) markerless systems (processing and inserting of virtual objects is without exact markers. Additional information is usually needed, for example, image and face recognition, GPS coordinates, etc.). The chapter contains also a description of mixed reality as an advanced computer user interface and the newest collaborative mixed reality.

Keywords: virtual reality, mixed reality, augmented reality, augmented virtuality, optical see-through systems, video see-through systems, mixed reality interface, collaborative mixed reality

1. Introduction

Mixed reality (MR) is the most advanced technology of today's virtual reality (VR) systems. It is the area of computer research dealing with a combination of real-world and computer-generated data. Computer-generated graphic objects are mixed into the real environment and vice versa in real time. Mixed reality, based on Azuma [1]:

- Combines real and virtual space
- Is interactive
- Is processed in real time
- Is registered in three dimensions

Mixed reality represents a combination of real and virtual worlds, where virtual data are inserted into the real environment or vice versa. The main function of mixed reality system is computer-based harmonization of real and virtual scene coordination systems and overlap of virtual and real images.

The virtual fixtures were the first mixed reality platform developed in 1992 at the Armstrong Laboratories of the USAF [2]. This project allowed virtual objects to overlap with the real environment in a direct user view. At present, mixed reality can arise using at least one of the following technologies: augmented reality (AR) and/or augmented virtuality (AV).

Mixed reality technologies give to users the chance to get a new experience. This solution, as already mentioned in classic VR systems, is particularly suitable for the presentation of design, urban, and architectural studies. It is a preview of a new form of visualization of real-world objects enhanced with virtual complementary information. A model can be created using 3D modeling tools, respectively, using direct export from, e.g., CAD tools, and they put into the real scene. The subsequent resulting scene of mixed reality can be created using some of the AR systems (marker or markerless). The correct placement of virtual objects in the scene is used either by markers or by other positional reference devices (e.g., GPS). Virtual objects together with the view of the real world create a mixed environment. They form a solution that brings a totally new form of computing resources usage overall in human-computer interfaces (HCI). In **Figure 1** a principle of the system of relations between the two areas/subjects is shown, and it cannot exist only on a computer but also on any device/system. For example, a TV remote controller has a user interface. This concept is valid also for mixed reality systems, but in this case (MR), it must be more natural and more interactive (one subject is human). Thus, MR can also be a good example of improving the interface for people with disabilities or for their therapy (see also **Figure 23**). A very nice example is a study described in the chapter "Using Augmented Reality Technology to Construct a Wood Furniture Sampling Platform for Designers and Sample makers to Narrow the Gap between Judgment and Prototype." The 3D printing output was included into mixed environment, and so limitations have appeared here. The form and state of sampling through innovative experimental methods were simulated. MR system design, aiming to quantify the objective data on furniture sampling on the shape, was presented, but because the size of the 3D printing was much smaller than the actual sampling size, the difference between the visual judgment of MR system users and the spatial shape was affected. This demonstrates the importance of the coordinate systems of the MR system components' coordination in terms of the interface's naturalness (see also **Figure 6**).

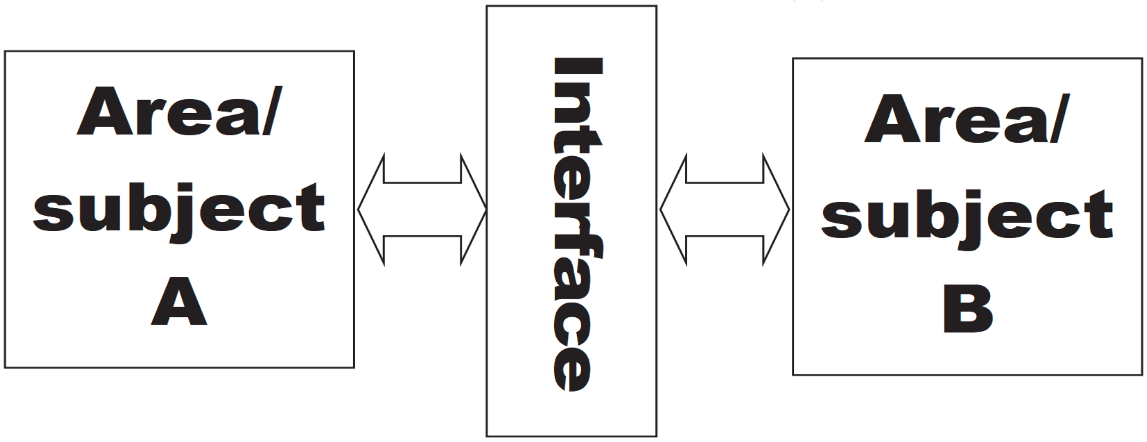

Figure 1.
Mixed reality as user interface concept.

The AR environment contains both real-world objects and virtual (synthesized) objects. For example, a user working with an AR system uses a display device (e.g., transparent display glasses or head-mounted display (HMD), monitor+camera combination), and he can see the real world combined with computer-generated (synthesized) objects displayed "as" on the surface of this world.

Augmented virtuality is similar to AR. Unlike AR, AV is the opposite approach. With AV systems, most of the displayed scene is virtual, and real objects are inserted into the scene. When a user is embedded in a scene, it is, like embedded, real objects, dynamically integrated into the AV system. It is possible to manipulate both, virtual and real objects in the scene, all in real time.

Both of these systems are quite similar, and both fall, as already mentioned, under the concept of mixed reality. Mixed reality includes both augmented reality and augmented virtuality. It is a system that attempts to combine the real world and the virtual world into a new environment and display, where physically existing objects and virtual (synthesized) objects coexist and interact with each other in real time. The relationship among mixed reality, augmented reality, augmented virtuality, and the real world is shown in **Figure 2**. An extended continuum by using of terms such as *real reality*, *amplified reality*, *mediated reality*, or *virtualized reality* (see chapter "Mixed reality in the presentation of industrial heritage development," **Figure 1**. Order of reality concepts ranging from reality to virtuality) is based on Milgram's continuum. Mediated reality is also included in Mann's classification.

In Mann's classification (**Figure 3**), the classification space is extended by mediality [4]. It means mediality in the form of mediation. The mediation in terms of this technology is an extended term encompassing certain objects of transferring visibility (visualization) to another format, i.e., transforming objects into a "media" form. And so, the mediation is understood as a process of transferring (transforming)

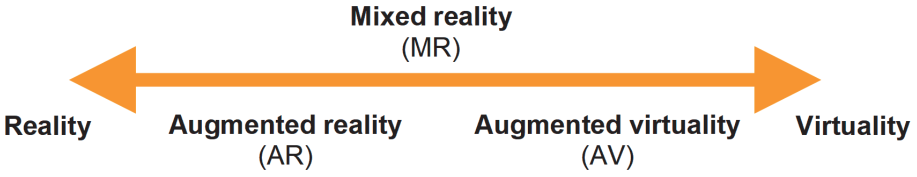

Figure 2.
Milgram's continuum between reality and virtuality [3].

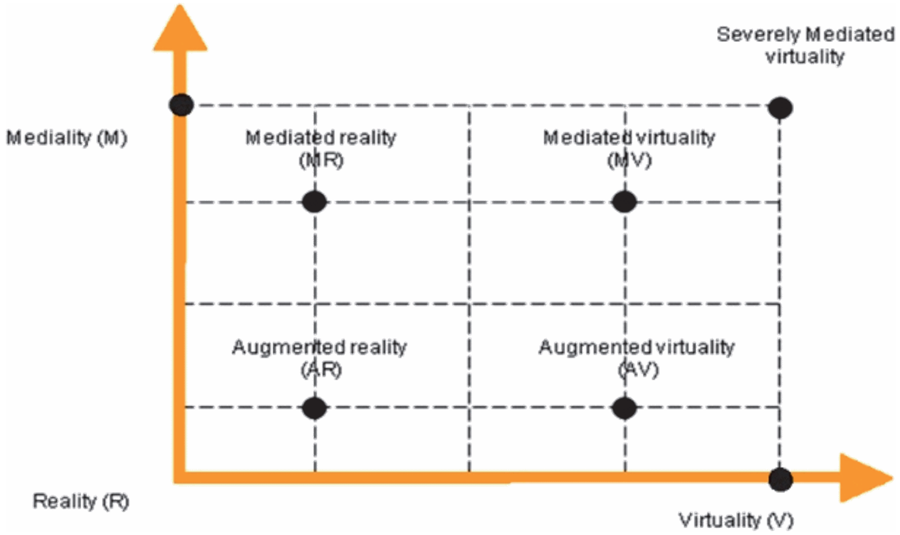

Figure 3.
Mann's classification of mixed reality systems (mediated reality continuum) [4].

data within the object creation or movement, including a set of transformations which allowed the transport of data for visibility (visualization). Overall, mediality is understood as an interactive interface, i.e. the environment of different worlds contact. It is, therefore, a measure of the possible interconnection between heterogeneous worlds using different forms of mediation (visibility, visualization).

Depending on how the user sees the mixed reality, these systems can be divided into two types:

- *Optical see-through systems*—the user can see the real world (reality) directly with the computer-generated (virtual) objects added (**Figure 4a**). These systems typically work with HMDs with transparent displays. Then, in **Figure 6** the R connection is not realized, and the real scene view is directly through this display.

- *Video see-through systems*—the real-world scene complemented by virtual objects is displayed to the user in a mediated manner, e.g., using the camera-display combination (**Figure 4b**).

There are two MR systems used to coupling virtual objects with the real world:

- *Marker systems*—special markers are placed in the real scene that are recognized and replaced by virtual objects during the runtime. QR codes or EAN codes can be used as markers, in addition to specialized markers.

- *Markerless (semi-markerless) systems*—systems without (special) markers—contrary to the marker AR, there is no need to have special markers in the real scene. GPS

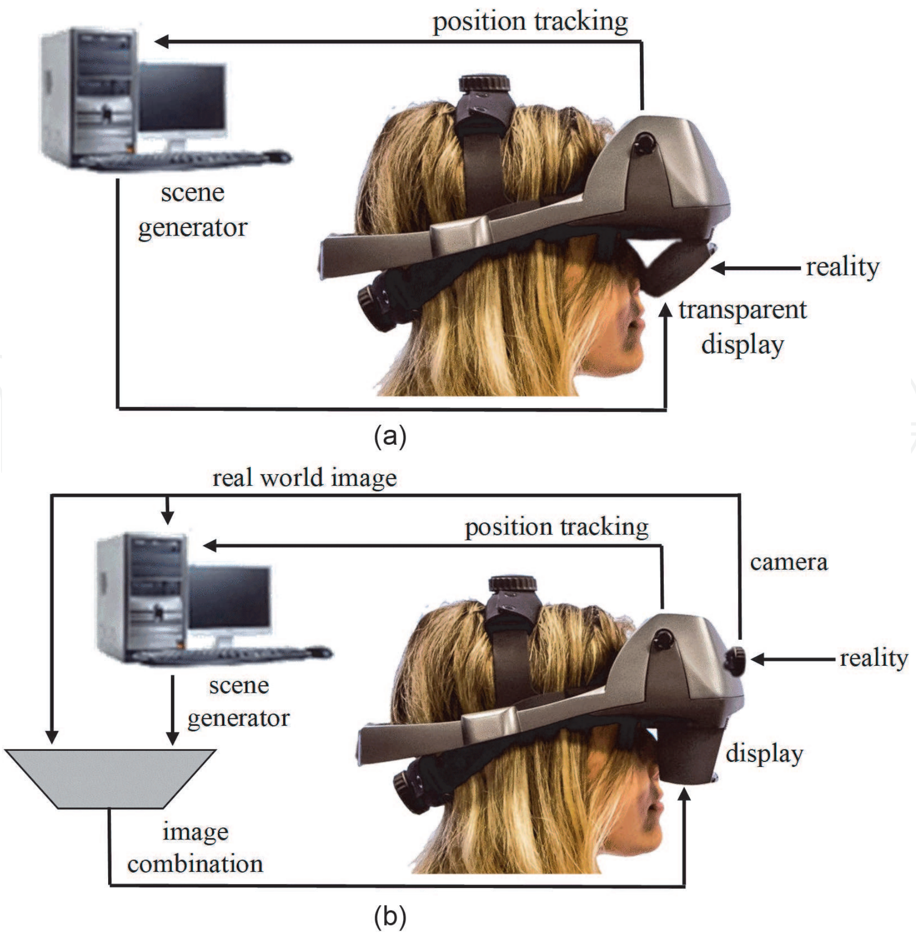

Figure 4.
Schematic representation of a mixed reality system optical see-through (a) and a video see-through systems (b).

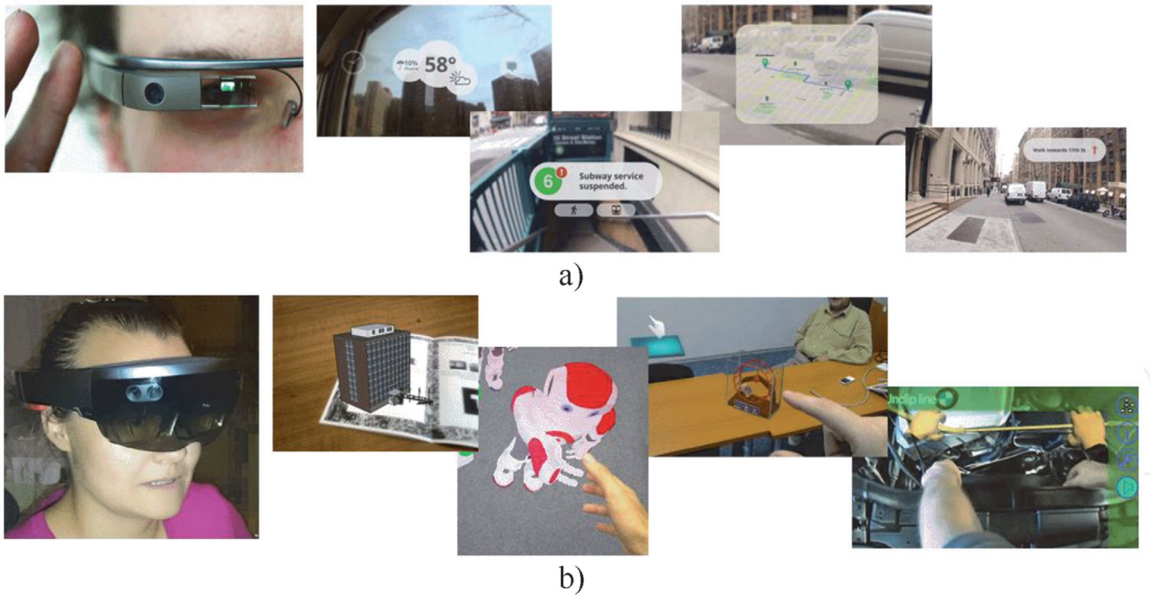

Figure 5.
Extended (a) and enhanced (b) mixed reality systems.

coordinates, Wi-Fi signal, camera output analysis (e.g. image recognition) and other means are used to place a virtual object into the real scene. In semi-markerless systems, real-world objects, naturally placed in the scene (e.g. a TV remote control, a cup or a book), are used as markers.

Depending on the area where the MR system is operated, MR systems are divided into:

- *Interior* MR systems
- *Exterior* MR systems
- *Combined* MR systems (both interior and exterior)

Depending on the geometric relation between the real world and virtual objects, MR systems can be divided into:

- *Extended (enriched) MR systems*—without direct geometric relationships of virtual objects with real world (**Figure 5a**, (discontinued Google glass are used only as an example))
- *Enhanced MR* systems—with geometric relationships of virtual objects with real world (**Figure 5b**)

Starting with **Figure 5b**, the examples presented in this chapter are results of the LIRKIS laboratory at the home institution of the authors (Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice).

2. MR system function

A standard virtual reality system attempts to fully immerse the user in a computer-generated environment. This environment is maintained by a system whose displaying part is provided by a computing system with the virtual world rendering graphical system. In order for the immersion to be effective, the user's mind and sometimes his body must identify with the visualized environment. This requires that the changes and movements made by the user in the real world correspond to the appropriate changes/movements in the provided virtual world. Because the user is looking at the virtual world, there is no natural connection between these two worlds, and therefore the connection (interface) must be established. The mixed reality system can be considered as a definitive immersive system. The user cannot be any more immersed in the real world. The goal is to bind the virtual image with the user view. This linkage is most critical for AR systems because we are (people) much more sensitive for visual inaccuracies than standard virtual reality systems. **Figure 6** shows the combination of displayed areas (coordinating systems) that must be realized in the mixed reality systems.

The camera realizes a perspective projection of the real 3D world into the 2D projection plane. The internal (focal length and lens curvature) and external (position, viewing direction, or other settings) of the device accurately determine what is displayed on the display. Virtual image generation is realized using a standard computer graphics system (e.g., based on OpenGL). Virtual objects are displayed in

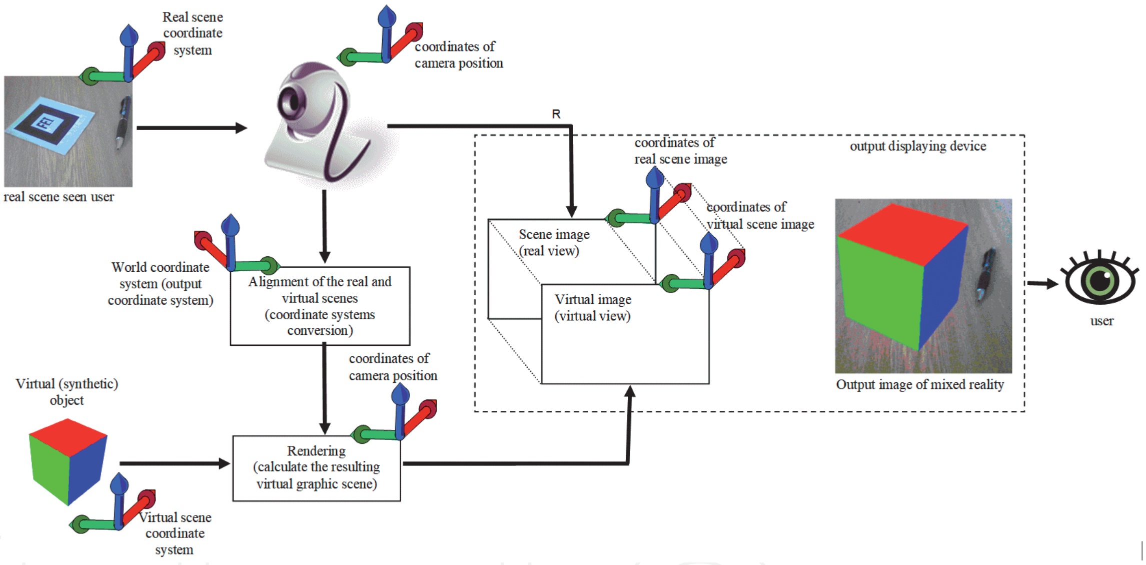
Real scene coordinate system
coordinates of camera position
R
real scene seen user
coordinates of real scene image
output displaying device
coordinates of virtual scene image
Scene image (real view)
Virtual image (virtual view)
World coordinate system (output coordinate system)
Alignment of the real and virtual scenes (coordinate systems conversion)
coordinates of camera position
Output image of mixed reality
user
Virtual (synthetic) object
Rendering (calculate the resulting virtual graphic scene)
Virtual scene coordinate system

Figure 6.
The combination of displayed areas (coordinating systems) in the mixed reality systems.

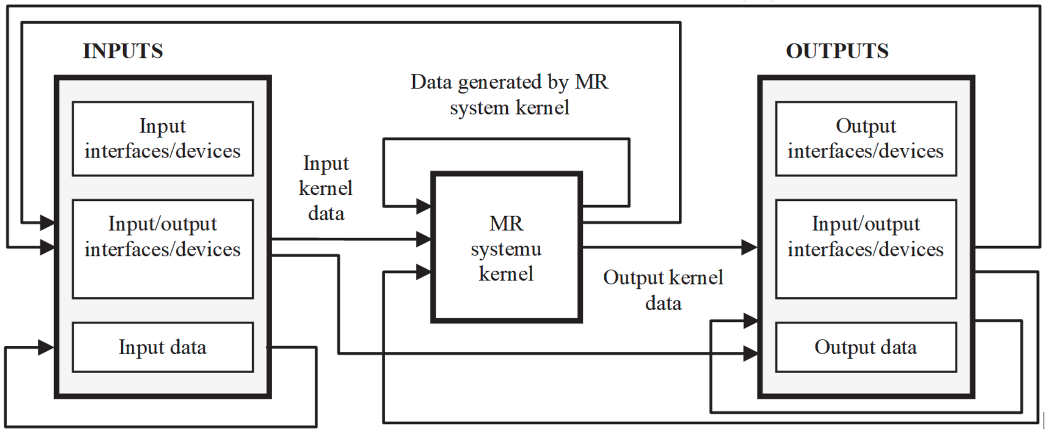
INPUTS
Input interfaces/devices
Input/output interfaces/devices
Input data
Data generated by MR system kernel
Input kernel data
MR systemu kernel
Output kernel data
OUTPUTS
Output interfaces/devices
Input/output interfaces/devices
Output data

Figure 7.
Schematic diagram of control/data flow in mixed reality systems.

a derived projection plane (screen). The graphics system requires information/data about the real scene image to render synthetic objects correctly. These information/data are applied to control of the virtual camera (computation of the inverse projection matrix) used to generate an image of virtual objects in the scene. This image is then merged with the real scene image to produce a mixed reality output image on the output display device.

The overall schematic way of implementing the MR system at the control and data flow level (**Figure 7**) is derived from the implementation of conventional VR systems. The biggest differences are at the input and output subsystem levels. This is mainly determined by the use of some special devices, e.g., transparent displays or gesture sensors. The abovementioned calculations of the inverse projection matrix, parts of image composition/combination, or image and possible marker recognition extend also the MR system kernel. In this case, the tracking subsystem is very important as described in the chapter "An interactive VR system for anatomy training" (**Figure 1**, Conceptual Diagram (Tracking module)).

3. Implementation of MR system with markers

Several stages are required in the process of implementation of AR technologies [5]. The first one concerns the preparation of virtual objects as 3D models. However, this can be performed by various technologies and principles. Therefore, the creation of 3D objects is possible through the following:

- 3D modeling tools and applications (for instance, a Trimble Sketchup).
- Utilization of 3D scanners.
- Modification of the existing 3D model.

In the second stage, the whole model is verified and performed to the required output format (OBJ, 3DS, GLTF, VRML, FBX, etc.). The type of output format depends on the engine and graphics library, which utilizes the AR application. The third stage contains the preparation of markers that are used for model placement into a physical environment. The fourth stage focuses on marker detection when the AR application is running. Then the proper visual output of the virtual object is performed. Detection of AR markers is conducted in real time by runtime processes that are responsible for visual output handling. Concerning the markerless MR system, the third and fourth stages are omitted and replaced by technology able to merge the real environment with included virtual objects.

The preparation of scenes purposed for mixed reality usage takes different technological scopes than AR. Even though the basis of AR is utilizing markers, there are still situations when some of them are out of detection range. In that case, the detection failure occurs. Unlike AR, mixed reality is more powerful and user-friendly which increase its usability for common usage. Utilizing depth-sensing to scan the surrounding physical environment is more effective in producing more enhanced visual content. All the virtual objects behave more naturally when they are placed in physical surroundings. Mixed reality devices also utilize depth-sensing to provide gestural interfaces for natural interaction. Mixing virtual objects and user's hands immerses human perception to manipulate virtual content more naturally. **Figure 8** contains a complete description of the whole process of creation MR scene as well as shows the basic structure of own created applications. Some steps are similar as in the case of a semi-markerless system (**Figure 12**).

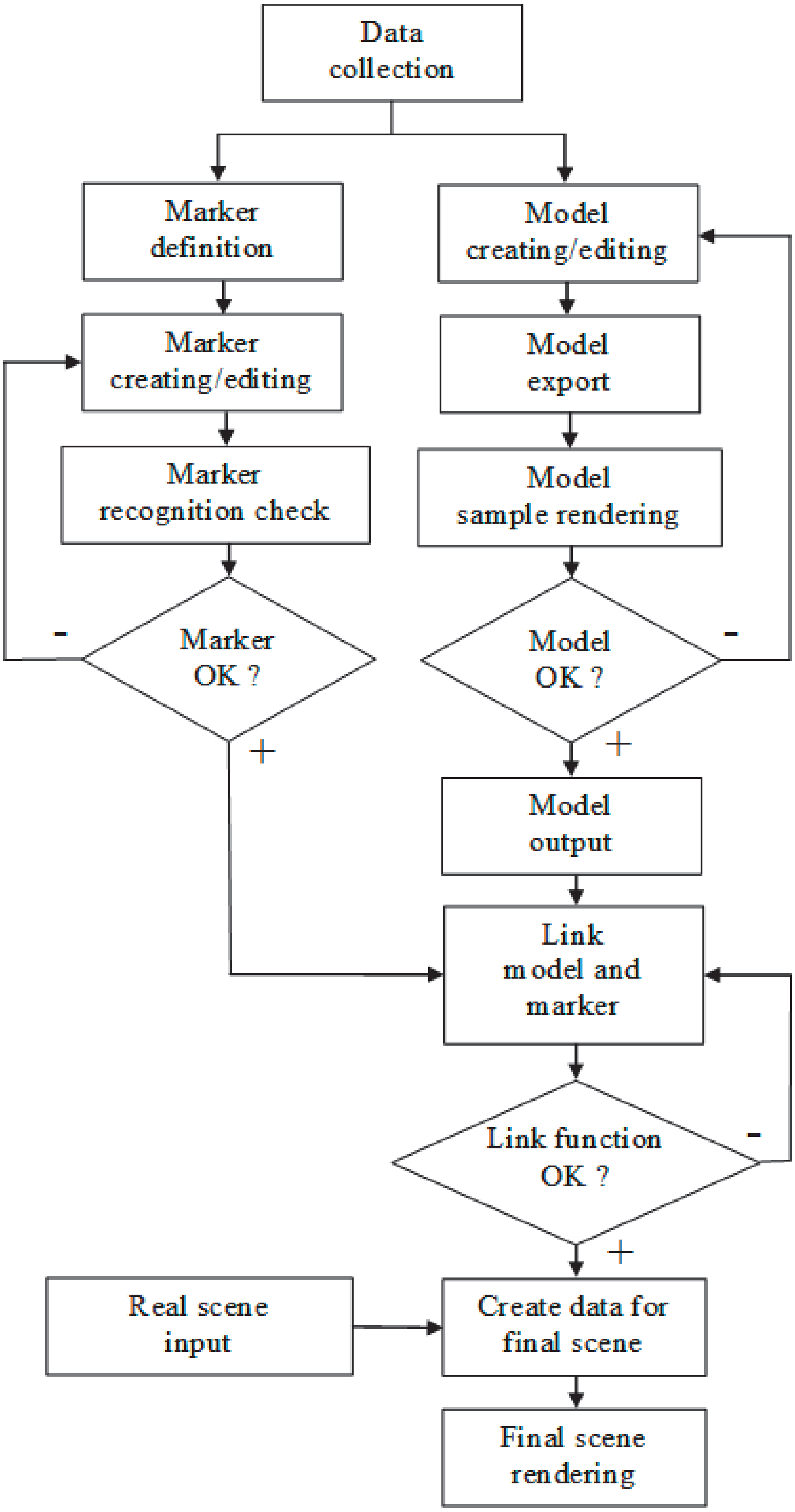

Figure 8.
Marked mixed reality creation process.

One of the problems of marker-based MR systems is marker design and size. The most important factors of correct recognition are marker complexity, camera resolution, scene lighting conditions and the distance between the camera and the marker. A bigger marker improves chances for recognition. It is advisable to

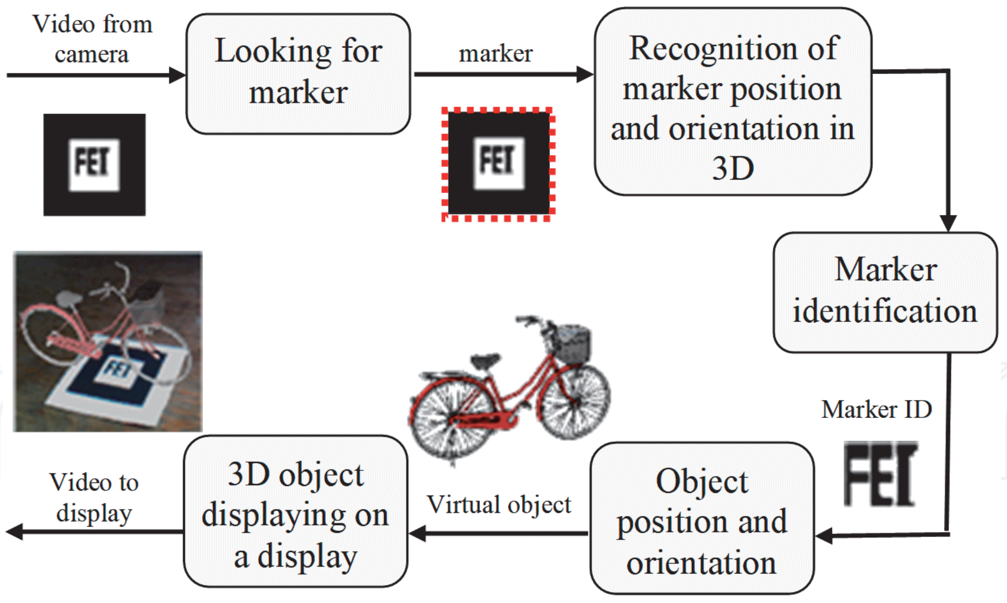
Video from camera
Looking for marker
marker
Recognition of marker position and orientation in 3D
Marker identification
Marker ID
FEI
Object position and orientation
Virtual object
3D object displaying on a display
Video to display
FEI

Figure 9.
Runtime process of marked mixed reality system.

use markers that contain combinations of larger areas with high contrast between them.

On top of already mentioned criteria, there are additional ones that have an effect on correct recognition of marker—the whole marker needs to be in the field of view of a camera; there is a problem with recognition if part of the marker is covered. Difficulties occur as well under low light conditions and when marker orientation toward the camera is not ideal. Too bright light source brings an additional set of problems as well as bright spots and reflections from the surface of the marker. The marker does not necessarily need to be printed on paper or sticker and surfaces with better contrast, and antireflective coating can be used. Another way to tackle problems with recognition is to print a marker visible under UV light, etc. The most used marked MR system is based on older ARToolKit software library (Software library for building AR applications created by Human Interface Technology Laboratory: http://www.hitl.washington.edu/artoolkit/), and schematic diagram of runtime process based on this library is shown in **Figure 9**. The one example of a typical AR Toolkit usage is presented also in the chapter "Augmented Reality as a new and innovative learning platform for the medical area" (see **Figure 1**. Image of a two-dimensional (2D) human heart placed in front of a camera where typical ARToolKit marker is used).

4. Mobile mixed reality implementation

Mobile mixed reality introduces an intelligent interface accessible for mobile devices. This technology originated outside the primary interest, for which the MR was invented [5]. Mobile MR can be performed by utilizing these technologies and services:

- Global positioning.
- Wireless communication.
- Location-based calculations.

- Location-based services.

- Mobile devices.

Each of the mentioned services and technologies provides localization of virtual objects and performs their proper visual output. Concerning mobile data services, the virtual object can be placed globally around the world without the limitation of geographical distances. The biggest challenge in mobile augmented reality is tracking and registration. Mixed reality applications include two separate components, which cover a whole process from setting markers and 3D models to producing visual output. The first component introduces a standalone application. Its main objective is to combine markers and 3D models into "datasets" and upload them to a server or networked storage. The second component contains a mobile application, which obtains datasets from the network and then renders whole 3D content. The overall design and functionality are described in **Figure 10**.

The standalone application can be written in C#. The mobile application (e.g., android app), however, is more complex. Usually, a software library support is needed. Two libraries working together can be used: *Vuforia* and *min3d* or a similar one. The first one (main part), Qualcomm AR/Vuforia (http://www.qualcomm.com/solutions/augmented-reality), is a library developed by Qualcomm Inc. company, especially for mobile devices. This library is meant for marker detection and simple 3D model rendering. The second one is meant to be a simple 3D engine, but in this case, it can be used solely only for 3D model rendering. Also, another library/framework can be used. The output is combined similar to **Figure 6**.

Because of the limited 3D model capabilities Vuforia has, the library will be modified so that it does no rendering at all, only marker recognition in the camera output. All rendering will be done by the 3D rendering library (min3d) based on the data it receives from Vuforia. The main disadvantage Vuforia library is the way to build markers for augmented reality. These markers must be made on the official site of the library.

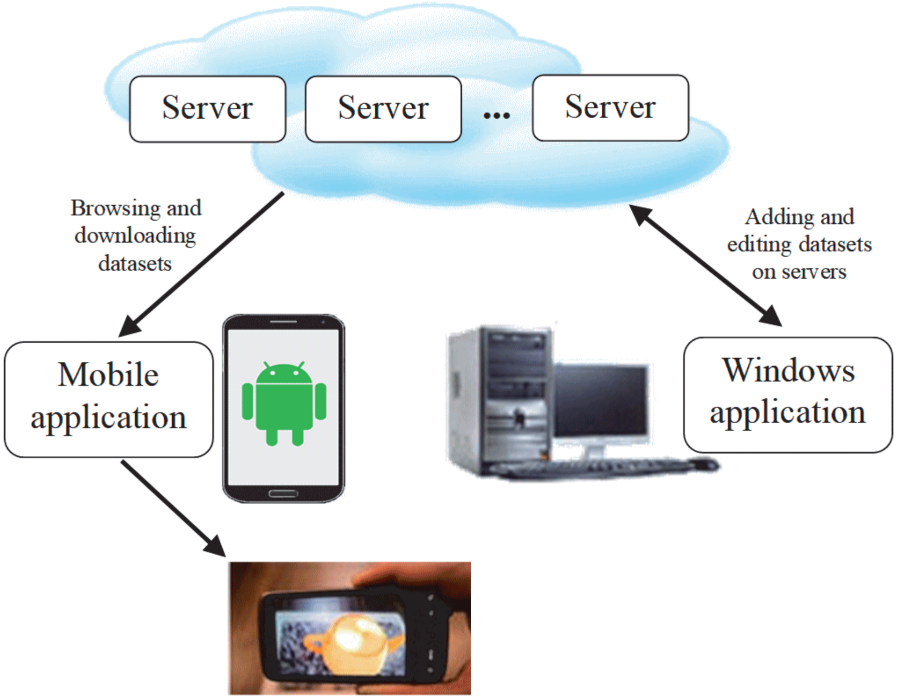
Server
Server
...
Server
Browsing and downloading datasets
Adding and editing datasets on servers
Mobile application
Windows application

Figure 10.
Mobile mixed reality application architecture.

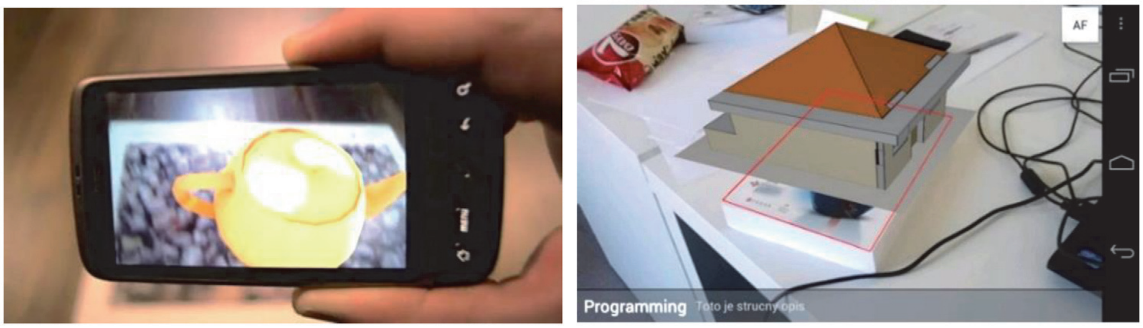

Figure 11.
Examples of the "augmented reality screen" on mobile (android) platform.

Then the augmented reality screen is the most important part of the application. It creates an augmented reality based on the dataset users choose. The resulting application is fully capable of creating an augmented reality, with the output displayed in **Figure 11**.

5. Implementation of markerless (semi-markerless) MR system

As it was already mentioned, it is more difficult to implement MR systems without exact markers (so-called semi-markerless and markerless systems). The whole process then uses objects that occur in the environment normally instead of artificial markers. It also utilizes other means, such as recognition of images, gestures or faces, depth cameras, 3D scanners, and GPS or Wi-Fi signal strength.

This technology can be divided into three types, which differ in the way the position and orientation of the inserted graphical entity are obtained:

1. By recognizing observed objects in the real environment, e.g., detection of points, edges, lines, etc.

2. By recognizing planar surfaces, e.g., texture recognition (semi-markerless systems)

3. Using information from another source, e.g. GPS

Regarding the first type, to be able to add a virtual object to a real environment (image), captured by a camera, we need to know the exact position of the virtual object. But the position changes when the camera is moved. In practice, this means that the virtual object remains fixed in the real image in the real environment and the look on it changes with the camera. The key part of this technology is environment tracking (scanning). This means that the system is always checking the position and orientation of the camera as well as detecting certain natural environmental clues (points, edges, etc.). Using these clues, we can add more graphical information to the image. And we know the position and orientation of the inserted virtual object. It is a computationally demanding process, considering that it should be computed in real time. It is appropriate to apply parallelization when implementing it.

The second type uses planar surface recognition. The planar surface may be a painting, a book cover, a photograph, a face, and so on. This technology is similar to the marker-based MR. However, it uses a specific rectangular planar surface (painting, photo, etc.) instead of an artificial marker. Various filters, as well as methods to identify significant points in the image, are used to recognize a texture

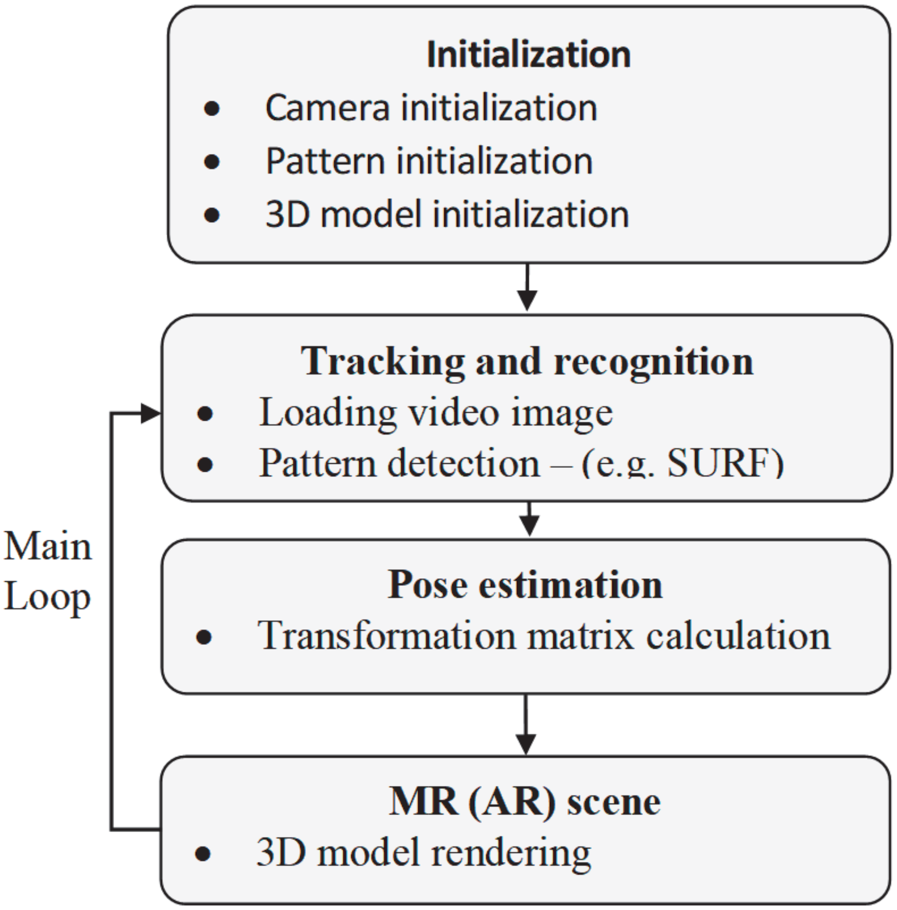

Figure 12.
The architecture of the semi-markerless system.

in the image. In this case, however, the computational demands of the application significantly increase, especially when detecting recognized shapes. How an MR system of this type works is shown in **Figure 12**. In this type of system, a learning phase is required. The learning phase involves scanning the environment for examples of objects we need to recognize and acquiring templates of these objects, e.g., in the form of their photographs.

The third type is used primarily in smartphones (see previous subchapter "*Mobile MR implementation*"). It uses the phone camera, which scans the place where the user is looking. Using GPS, the system will detect where the user is and which points he has in his surroundings. The digital compass of the smartphone is used to determine the direction in which he is looking. The use of these features of the smartphone (camera, digital compass, GPS) allows creating MR applications.

The principle of creating an MR without exact markers is similar to creating an MR with exact markers (**Figures 8** and **9**). However, there is a significant difference in the method of recognizing the original and positioning it in the real scene image.

How markerless (semi-markerless) MR works can be, on the basis of **Figure 12**, described by the following steps:

1. After initialization, the camera constantly captures the real scene and sends the video to the computing system for processing.

2. The software processes the captured image by frame and searches for the pattern(s)/object(s) in the image using the selected detection method.

3. The position and orientation of the object/s (pattern) are computed after it is recognized (computer vision area).

4. After the position and orientation are known, the virtual object model is placed at the position.

5. The user sees the real scene, as captured by the camera (*video see-through systems*) or as seen through the transparent display (*optical see-through systems*), with the virtual object added.

Steps 2 and 3 are essential and the most demanding ones. The most commonly used methods for image recognition are based on *SIFT* and *SURF* algorithms.

- *SIFT* means *scale invariant feature transform*. It is named after the principle it uses—it transforms images to coordinates independent from the scale. It is one of the more recent methods for significant point detection. In [6], David G. Lowe says that the points found do not depend on scale, rotation, affine deformations, noise, and illumination changes.

- *SURF (speeded-up robust features)* is a more recent method, inspired by SIFT. The description of an image, generated by this method [7], is invariant to image rotation and distance between the camera and the described object. SURF is used in many computer vision applications, for example, 2D and 3D scene reconstruction, image classification, and fast image description creation.

The implementation of semi-(markerless) mixed reality consists of four main components: *initialization*, *tracking and recognition*, *pose estimation*, and *MR scene* [8]. The architecture of the semi-markerless mixed reality system is shown in **Figure 12**. The implementation of this system required two additional platform-dependent software packages. The first one was *NyARToolkit* (https://nyatla.jp/nyartoolkit/wp/) with the core of the mixed reality construction and also an implementation of mathematical calculations used for determination of the pattern/object position. The second one was the *Emgu.CV* software library (http://www.emgu.com), which provides the already mentioned SURF method implementation for the detection of patterns/objects in the image.

The component *initialization* sets some parameters of the camera, pattern/object, and 3D object.

The component *tracking and recognition* recognizes the pattern/object from the image captured by the camera. This step can use the SURF method, e.g., from the software library Emgu.CV. This method describes the image by using descriptors. The description with the descriptors generated by this method is invariant to rotation and camera distance from the object being described. Interest points obtained by this method are shown in **Figure 14**. 3D scanning technology and followed recognition can be used also in this component. However, a detailed description of this method goes beyond the scope of this chapter.

The component *pose estimation* calculates the transformation matrix, for the establishment of the three-dimension coordinates on the pattern/object. For the calculation (based on [9]) itself, it is necessary to know the projection matrix, which is obtained by camera calibration. The most important part of the calculation is to obtain a transformation matrix that determines the location of the 3D virtual graphic object into 3D space. Placing the virtual model into the real world is needed to determine the parameters of the transformation matrix. In case we have a pattern (square/rectangle) as shown in **Figure 13**, determination of the transformation matrix parameters is as follows (1) and (2):

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} & V_{13} & W_x \\ V_{21} & V_{22} & V_{23} & W_y \\ V_{31} & V_{32} & V_{33} & W_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} V_{3\times3} & W_{3\times1} \\ 000 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = T_{\text{cm}} \cdot \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \tag{2}$$

T_{cm} (transformation from pattern coordinates to camera coordinates) is obtained by analyzing the input image. This transformation matrix consists of the rotation matrix ($V_{3\times3}$) and the translation matrix ($W_{3\times3}$). Two parallel patterns edges (margins) are reflected in the image. Coordinates of these edges correspond to the equations of lines (3):

$$\begin{aligned} l_1 &: a_1x + b_1y + c_1 = 0 \\ l_2 &: a_2x + b_2y + c_2 = 0 \end{aligned} \tag{3}$$

The determination of the line parameters can be calculated in several ways. One of them is a calculation of parameters, if we know at least two points that lie on this line. Because pattern/object has a square or rectangle shape, we can obtain coordinates of its four vertices in the screen coordinate system. These coordinates are obtained using the SURF method after pattern/object recognition in the video image. Denote the pattern as a rectangle *ABCD* (**Figure 14**). Edges *AB* and *CD* are parallel. Corresponding equations for these edges are equations of lines l_1 and l_2 (3). Also, the edges *BC* and *DA* are parallel and their equations are l_3 and l_4.

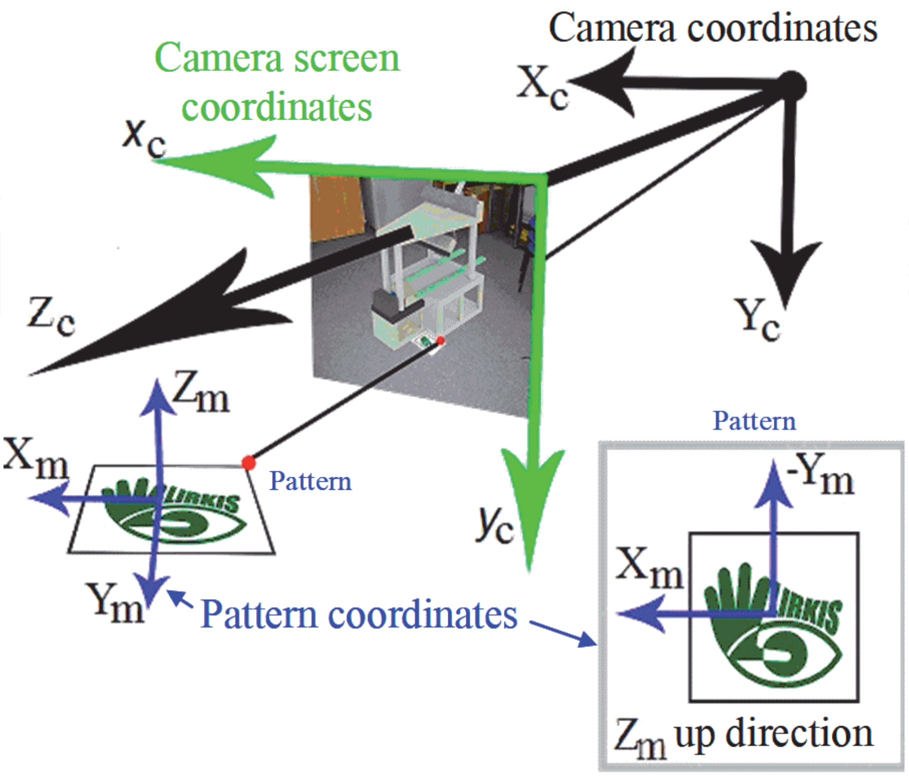

Figure 13.
The relationship between pattern coordinates and the camera coordinates.

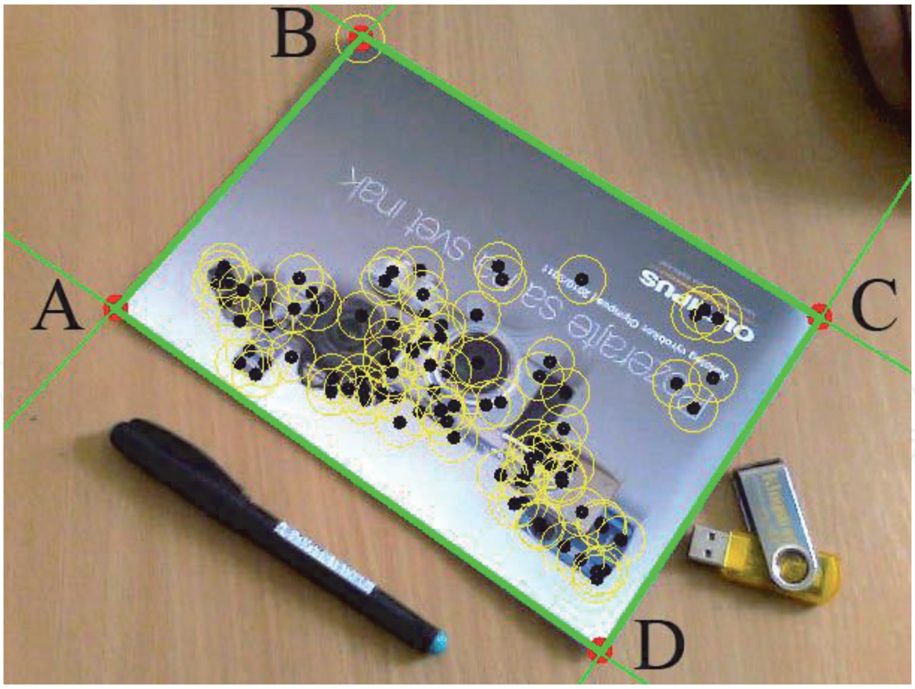

Figure 14.
Rectangle ABCD and interest points obtained by SURF method.

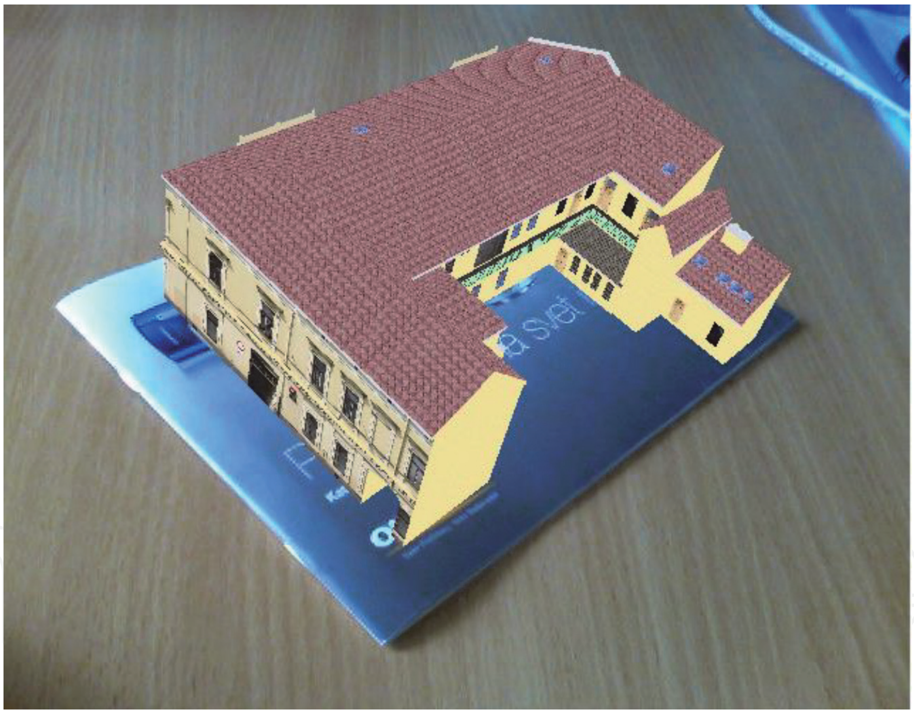

Figure 15.
Semi-markerless augmented system. The virtual model is displayed in the real world.

Determination of line parameters l_1:

1. Finding of direction vector line

$$\vec{u} = |\mathrm{AB}|, \mathrm{A}[a_1, a_2], \mathrm{B}[b_1, b_2].$$

2. $u_1 = b_1 - a_1; u_2 = b_2 - a_2.$

3. $\vec{u} = (u_1, u_2).$

4. Determination of the vector that is perpendicular to it: $\vec{n} = (u_2, -u_1)$.

5. Substitution of the values into the general equation of the line $ax + by + \mathrm{c} = 0$:

$$u_2x - u_1y + c = 0.$$

6. Substitution of the values x and y for the point that lies on a line such as coordinates of point B and computation of the parameter c.

In a similar way, the general equations of lines l_2, l_3, and l_4 are obtained. The next procedure is to calculate the rotation and translation part of the transformation matrix.

The last component *MR scene* displays the virtual model in the real world. To view mixed reality, an appropriate rendering core can be used. The example result is shown in **Figure 15**.

6. Mixed reality as user interface and gesture recognition

Gestural interfaces offer various features to provide hand tracking for nonverbal interaction [10]. In the mixed reality, hands are the most effective tools that can be used for natural hand-object manipulation. Unlike touch interfaces, there is an opportunity to work with a variety of gestures and transform their semantics to specific commands. Gesture-based interfaces give users the freedom to interact without any limitation than using contact VR controllers.

Considering human-computer interaction (HCI), gesture recognition is performed by a digital system that senses users' handshapes and responds to them [11]. Handshapes are equal to visual patterns, which are recognizable in real time. Nowadays, there are several technologies that can provide full hand tracking.

The *Microsoft HoloLens* (MS HoloLens) introduces an all-in-one head-mounted display, which supports the complete head and hand tracking. In contrast to other MR systems, the MS HoloLens can provide two-handed gestures to ensure more intuitive interaction [12]. The gesture recognition utilizes an infrared depth camera which senses the reflection of the user's hands [13].

The similar technology as MS HoloLens is Microsoft Kinect (MS Kinect), which provides motion sensing of the human's rigid body and hands [14]. The gesture recognition and body tracking utilize the same principles based on the depth sensor including an infrared laser projector. In contrast to MS HoloLens, the MS Kinect can sense multiple persons concurrently, who can interact together [15].

In general, mixed reality focuses on gesture recognition to intent powerful and natural HCI. The utilization of IR sensors proves excellent results in development and research [16]. One of the specific systems is *VirtualTouch*. The system supports human-object interaction [17], where virtual objects are merged into physical ones. The user operates with a physically based object which is wrapped by its virtual entity.

In mixed reality, gestures can be utilized to perform a single event or continual activity. The majority of gesture recognition considers two categories that consider gestures duration:

- Static gestures (considered as events executed in the shortest time intervals, **Figure 16**)

- Dynamic gestures (considered as an activity with longer time duration, **Figure 17**)

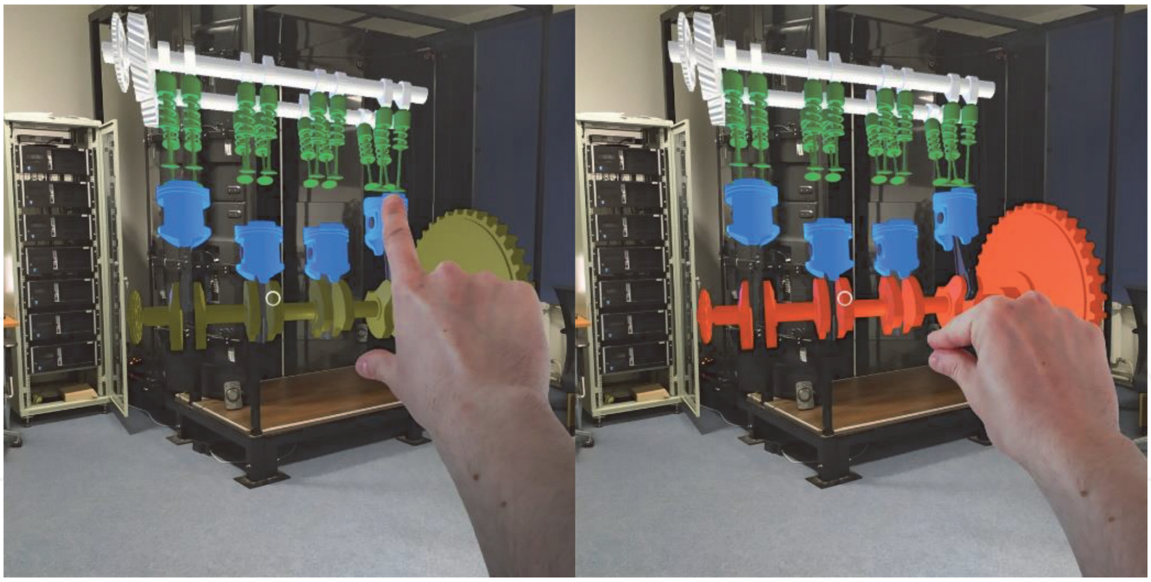

Figure 16.
Clicking on hologram, static gesture utilization.

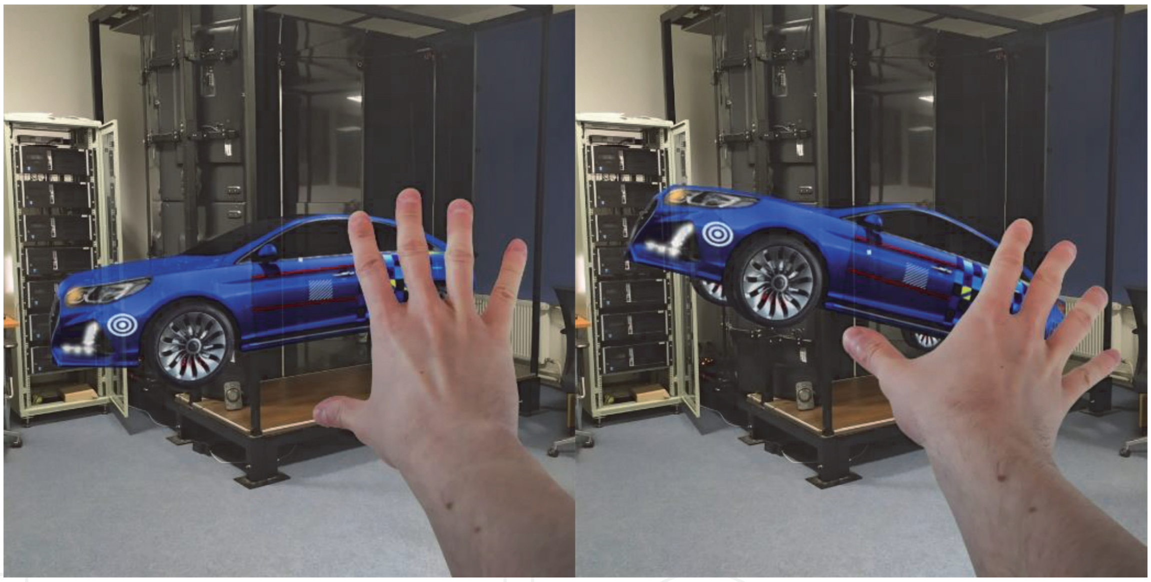

Figure 17.
Continuous hologram manipulation by a hand, dynamic gesture utilization.

6.1 Static gestures

The recognition of static hand gestures (**Figure 18**) in mixed reality uses the identification of hand poses in a stream of image frames [18]. The static gesture represents an event executed in the shortest time intervals [19].

Gestural interfaces based on static gesture recognition include several stages to process gesture inputs. The first stage concerns hand tracking technology able to sense human hand in real time. This is usually supported by depth sensors or infrared cameras. In the second stage, the image sequence is performed. The hand detection obtains a hand posture from the image sequence. Using a variety of detection techniques [20] can filter different hand poses. In the third stage, the image segmentation preprocessing is provided. Then all of the detected hand regions are filled by contrasting colors and sharpened on boundaries. The final hand boundary representation is necessary for gesture recognition [21]. In the fifth stage, the obtained gesture is compared with records from gesture datasets. If the

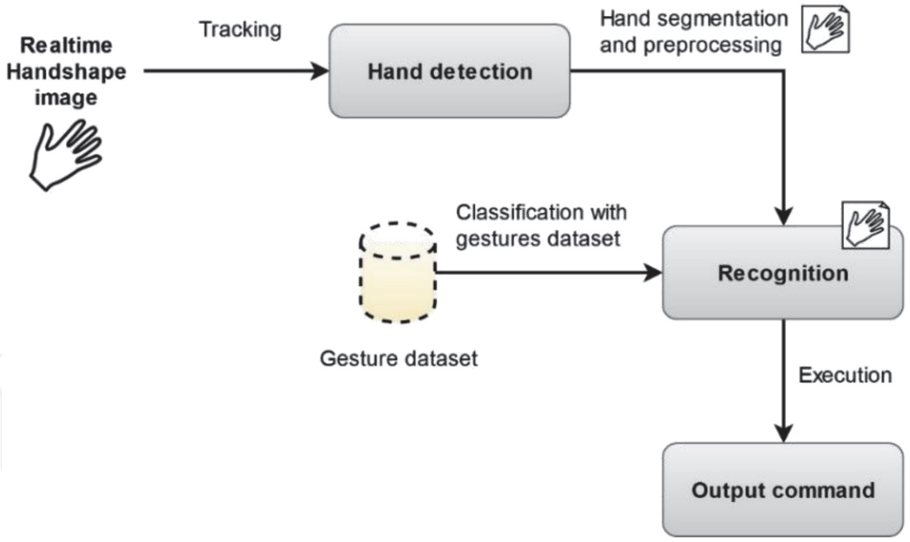

Figure 18.
Detection of static hand gesture interaction in real time.

classification of detected gesture is similar to its dataset record, then recognition is successful. In the final stage, the gesture is executed into the output command.

The advantage of static gesture recognition concerns the storage of gestural dataset records in simple readable structures such as images and text files. On the other hand, the preparation of new gestures requires the preparation of large dataset records.

6.2 Dynamic gestures

Continuous dynamic gestures (**Figure 19**) represent the activity sensed over a long time during which the movement of the human hand or limb is carried on [22]. The reason for utilizing continuous gestures in mixed reality refers to the interaction based on continuous manipulation of a virtual object. In contrast to static gestures, the preparation of dynamic gestures utilizes diverse principles in tracking [23]. While static gestures contain detection of hand posture, dynamic gestures equip motion tracking. The motion tracking performs real-time detection of the user's hands and limbs concurrently.

Most mixed reality systems support dynamic gestures to provide natural interaction. During the continual activity, the user can pick up virtual objects and manipulates them. This activity is triggered by static gestures that manage the beginning and terminating of dynamic gestures. As shown in **Figure 20**, before the

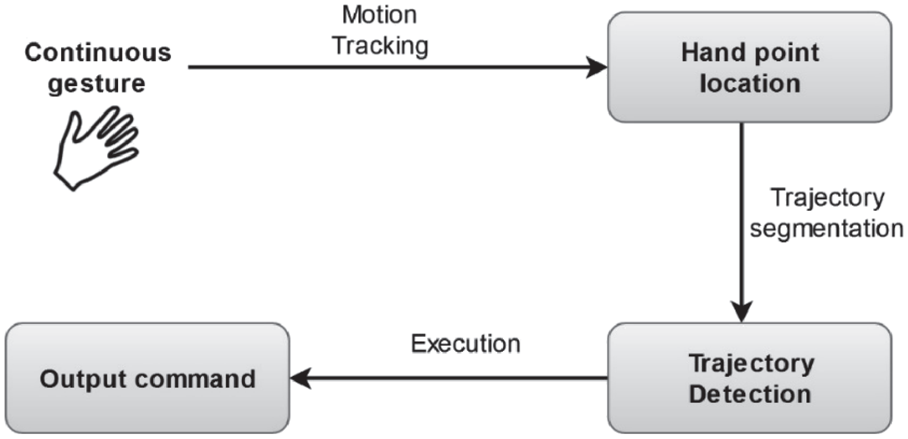

Figure 19.
Performing continuous dynamic gesture recognition.

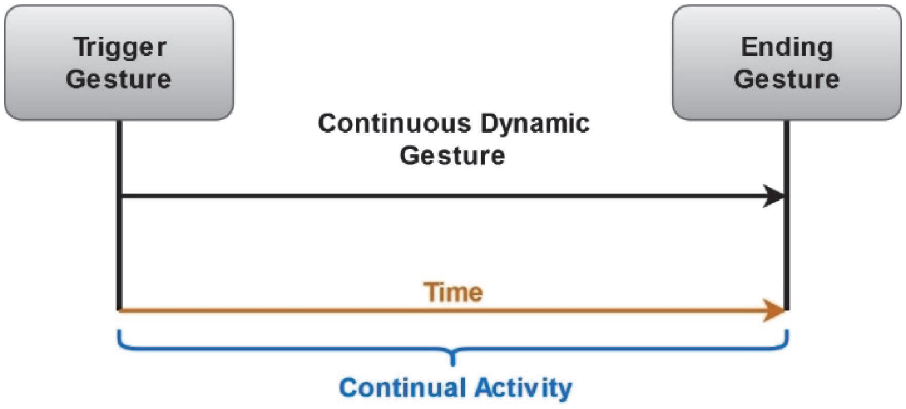

Figure 20.
Performing continuous dynamic gesture recognition.

activity starts, the trigger gesture is obtained. The whole activity (dynamic gesture) can last over a long time, while the user interacts with virtual content. After the activity fulfills, the ending gesture terminates the action.

7. Mixed reality speech recognition

The human speech represents the most common form of everyday communication [24]. In terms of human communication, extending mixed reality with speech recognition has an effective approach to provide multimodal interfaces. Through voice commands, the user can naturally communicate with the system [25]. This kind of interface frees the user from the touch or haptics interaction. Speech commands can be helpful in situations when users perform activities that engage their hands. The uniformity of speech recognition interfaces results in excellent usage on different platforms. Nowadays, mixed reality applications are utilizing speech interfaces in fields of education, research, medicine, and industry (**Figure 21**).

The whole process of speech recognition includes four stages which concern the following [26]:

- Analysis of speech inputs

- Feature extraction

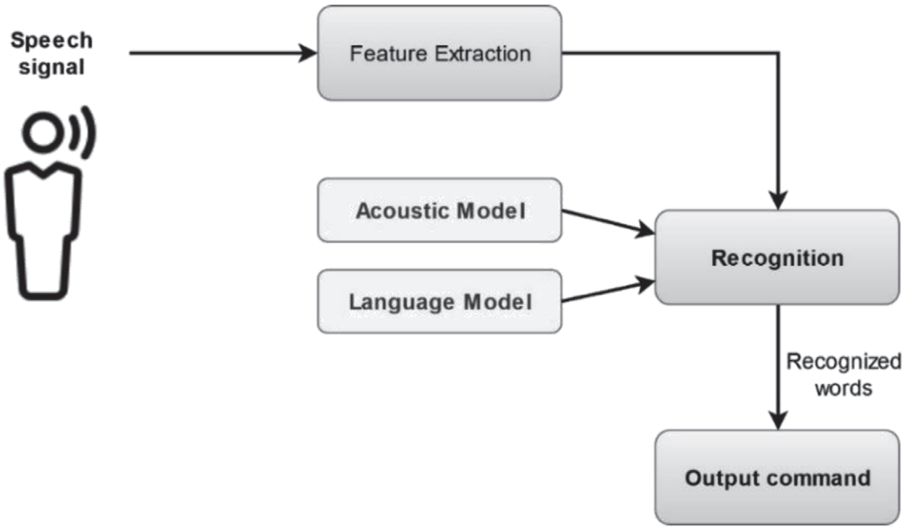

Figure 21.
Performing speech recognition.

- Speech recognition
- Decoding output command

7.1 Analysis of speech inputs

In the first stage, the system obtains speech inputs. The speech input can include one or even several words. After the speech input is recorded, it is important to convert its representation into the analog signal.

7.2 Feature extraction

The speech input can contain surrounding noise that affects the purity of speaking voice. This step focuses on extracting two waveforms from the input, the whole speech, and environmental sounds. The speech input is purified using various techniques based on spoken context, pitch and variation, duration, and frequency of speaking. Most of the mixed reality systems utilize the artificial intelligence components that provide automated feature extraction in short time intervals.

7.3 Speech recognition

This stage concerns the modeling techniques by using the acoustic and language model [27] to identify words in the speech input. The acoustic model works with audio records and process statistics of every spoken word to recognize syntax. The language model recognizes the semantics resulted from the speech input and detects the language in which the word is spoken. After performing speech identification, the final words are formed.

7.4 Decoding output commands

After finishing word recognition, the output command is performed. Each of the commands can perform various functions according to final use. Their functionality is fully unlimited. The speech recognition in mixed reality commonly prefers shorter speech inputs that are more effective than sentences. One-word commands are more specific and user-friendly.

8. Collaborative mixed reality

Mixed reality increases users' experiences utilizing gestural and speech recognition. This feature becomes useful for providing collaborative environments with multiuser interaction. Unlike other collaboration systems, collaborative mixed reality (CMR) offers a virtual and physical environment, where members can interact together. In fact, there are many systems designed for CMR purposes.

The CoVAR [28] introduces a remote collaborative system supporting VR and MR technologies. Participants can collaborate within the same local real-world environment or remotely. In the locally based collaboration, the MR user captures the surrounding physical space and shares its 3D model with other VR users. The remote collaboration utilizes the same principles but also a network to share a collaborative environment over long distances. The whole system primarily utilizes MS HoloLens for MR and HTC Vive for VR usage. In the case of interaction, the system inputs are formed to support head gaze, eye gaze, and hand gestures. The head gaze equips technologies included in VR and MR devices concerning the

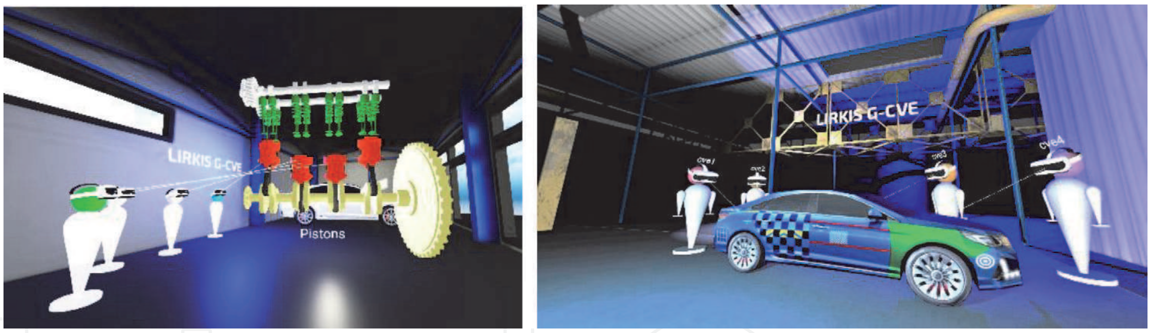

Figure 22.
An example of virtual collaborative environment with multiple avatars.

spatial mapping and head tracking movement. The eye gaze is supported by the Pupil Labs system, which tracks eye movement to ensure eye to object interaction. Gesture input is supported by hand tracking, for which MS HoloLens (in MR usage) and LeapMotion (in VR usage) are responsible.

The next of CMR systems called Vishnu [29] is concerning the mediation of virtual and real environments for remote guiding on a global scale. The system prepares separate visual outputs for MR and VR platforms. The whole collaboration focuses only on the objects that are captured by the MR side. The MR creates a real-time 3D scan and shares it with the VR side. The VR participant is able to manipulate a 3D scan and also can work together with the MR participant. The technological scope of the Vishnu includes hand tracking (OptiTrack and Kinect) and video-see mode through Oculus Rift stereo cameras for MR usage.

Another system [30] related to remote guiding through collaborative mixed reality utilizes 3D point cloud data. Two collaborators, the local worker, and remote helper can operate in a commonly shared environment. Both are using the same head-mounted technology (Oculus Rift DK2). The local worker captures his workspace through Oculus stereo cameras and distributes real-time visual output to the remote helper. The hands of the remote helper are captured by a depth sensor continuously. Their 3D point cloud overlays the visual output of the local worker even if it necessary to guide him.

The next point cloud collaboration [31] focuses on remote Telepresence where MR and VR are used to engage physically presented (on-site users) and remotely shared users (remote users) in one shared space. The on-site users are physically available in the same physical environment, while the remote users are connected over the network and presented by their 3D point clouds. The system affords interaction between all participants through high-res point clouds that include realistic bodies. All point clouds are captured by depth-sensing through Kinect V1 and V2. The interaction is performed by a gestural interface equipped with free-hand tracking through MS HoloLens and Leap Motion.

The LIRKIS G-CVE [32] introduces global collaborative virtual environments that are fully compatible with mixed reality usage (**Figure 22**). Unlike other collaborative mixed reality software and systems, the LIRKIS G-CVE is accessible through web browsers that ensure cross-platform support for a variety of VR, MR, and AR devices. All collaborative environments are distributed over the network. The system includes several interfaces, which enhance user interaction. There are gesture recognition, haptic interaction, and voice commands. The haptic interaction utilizes VR controllers equipped with three and six degrees of freedom. These immerse participants to interact more naturally and improve object manipulation. Gesture interface offers an intuitive object manipulation through MS HoloLens as grabbing, pulling, throwing, and stretching 3D object. These are currently limited to using only one hand than both. LIRIS project used MR and MS HoloLens for rehabilitation

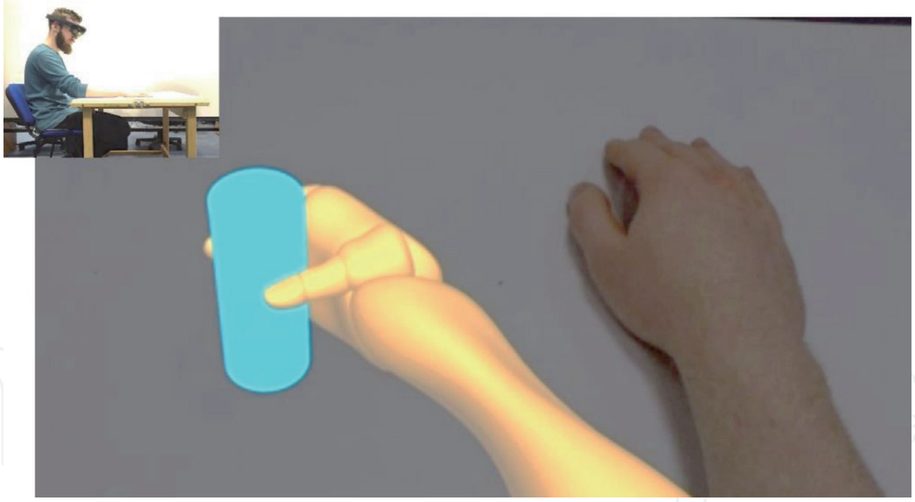

Figure 23.
An example of a patient's view in rehabilitation process using MR.

of patients after stroke, and training of movement of their hand is also very important. A patient uses MS HoloLens, and he can see real hand and also phantom virtual hand with appropriate movement. Then he can try to perform the suggested movements. An example of a patient's view is illustrated in the **Figure 23**.

The voice commands perform multimodal user inputs when utilizing other interaction techniques. Interacting through voice is limited to simple commands that are responsible for simple operations (enable and disable functions, hiding and showing 3D objects).

9. Mixed reality and SMART environment simulation

Building a SMART household without testing and implementing it into real operation is complicated and can be very costly. Therefore, simulators are created. The study [33] identified areas in which smart intelligence simulation research is being conducted. The study [33] shows an overview of some simulation tools analyzed for the SMART household. **Figure 24** shows the view from a created simulator of a SMART environment using freeware technologies such as Blender, Python, and JavaScript. The program serves to visualize smart home simulation with few basic appliances, which are used to present the way the simulator works. These appliances can be controlled using the control panel or with a direct approach using clicks and context menu. The control panel sets the profiles for appliances' statuses. It is possible to move freely in the household and interact with the appliances.

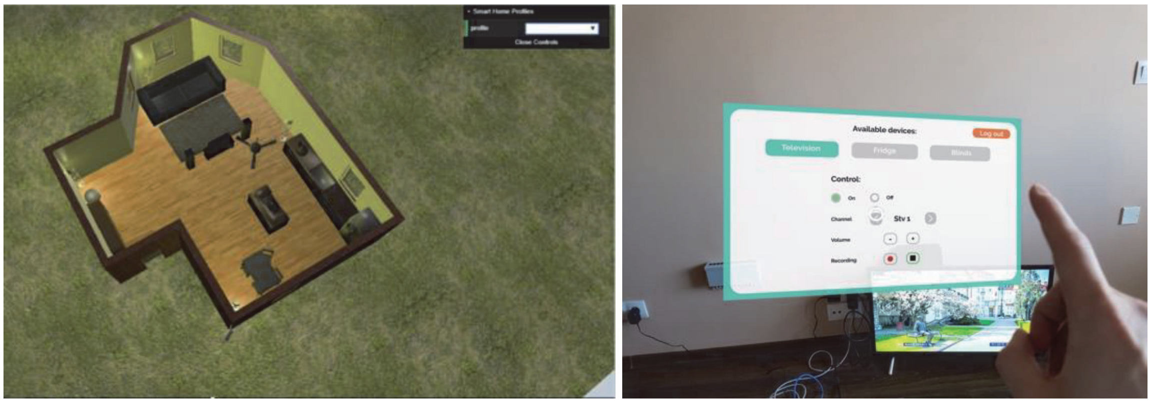

Figure 24.
Simulation model of SMART household (left) and real SMART household user interface control (right).

The user interface consists of a scene containing the model itself with appliances and other functional and nonfunctional object models. This simulation model and its smart appliances can be visualized as part of the mixed reality, using Microsoft HoloLens or other data helmets that can run a web browser. In this mode, the user can freely move and control appliances, such as turning on/off the television, lights, sunblind, etc. Users can also choose or modify one of the existing presets. Choosing presets, all appliances will set appropriate states based on the selected profile. For example, choosing "away from home" will turn off lights and TV and lock the doors. In such simulated environment, more users can collaborate because all requests and responses are done on the backend server and all users have actual data about simulated appliances states. This interface is also suitable for controlling households with handicapped people.

10. Conclusion

Mixed reality research is progressing quite well, although it requires significant financial resources. On the other hand, this technology offers a very immersive experience for its users. Mixed reality allows to bring gaming, education, training, and presentation of various kinds of designs up to an entirely new level. It represents a new form of visualization of real objects, extended with virtual information. Models can be created using 3D modeling tools, including CAD software, and inserted to a real scene. A mixed reality scene can be then created using one of the available augmented reality systems. The correct placement of virtual models inside a scene is ensured either by markers or by a combination of recognizable objects from the real environment and additional information from other sources, such as positioning systems. Together they create a solution that brings a new form of computing resource utilization.

Acknowledgements

This work has been supported by the APVV grant no. APVV-16-0202 "Enhancing cognition and motor rehabilitation using mixed reality" and by the KEGA grant No. 035TUKE-4/2019: "Virtual-reality technologies and handicapped people education."

Author details

Branislav Sobota*, Štefan Korečko, Marián Hudák and Martin Sivý
Technical University of Košice, Slovakia

*Address all correspondence to: branislav.sobota@tuke.sk

References

[1] Azuma R. A survey of augmented reality. Presence Teleoperators and Virtual Environments. 1997;**6**(4): 355-385

[2] Rosenberg LB. The Use of Virtual Fixtures As Perceptual Overlays to Enhance Operator Performance in Remote Environments; Technical Report AL-TR-0089. OH: USAF Armstrong Laboratory, Wright-Patterson AFB; 1992

[3] Milgram P, Kishino AF. Taxonomy of mixed reality visual displays. IEICE Transactions on Information and Systems. 2013:1321-1329

[4] Mann S. Campus Canada, ISSN 0823-4531; Feb–Mar 1985, p. 55; Apr–May 1986, pp. 58–59; Sep–Oct 1986, p. 72

[5] Sobota B, Korečko Š, Hrozek F. Mobile Mixed Reality; ICETA 2013: 11th IEEE International Conference on Emerging eLearning Technologies and Applications: Proceedings; 24–25 October 2013; Stary Smokovec, Slovakia. Danvers: IEEE; 2013. pp. 355-358. ISBN 978-1-4799-2161-4

[6] Lowe DG. Distinctive Image Features from Scale-Invariant Keypoints. Computer Science Department, University of British Columbia; 2004

[7] Tuytelaars BT, Van Gool L. Speed up robust features. European Conference on Computer Vision. 2006;**1**:404-417

[8] Varga M. Markerless augmented reality using SURF method; SCYR 2012. In: Proceedings from Conference: 12th Scientific Conference of Young Researchers; 15 May 2012, Herľany, Slovakia. Košice: TU; 2012. pp. 173-176. ISBN 978-80-553-0943-9

[9] Kato H, Bilinghurst M. Marker Tracking and HMD Calibration for a Video based Augmented Reality Conferencing System, Iwar. IEEE Computer Society; 1999. p. 85

[10] Chang YS, Nuernberger B, Luan B, Höllerer T. Evaluating gesture-based augmented reality annotation. In: 2017 IEEE Symposium on 3D User Interfaces (3DUI). IEEE; 2017. pp. 182-185

[11] Song Y, Zhou N, Sun Q, Gai W, Liu J, Bian Y, et al. Mixed reality storytelling environments based on tangible user interface: Take origami as an example. In: 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). IEEE; 2019. pp. 1167-1168

[12] Chaconas N, Höllerer T. An evaluation of bimanual gestures on the Microsoft HoloLens. In: 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). IEEE; 2018. pp. 1-8

[13] Xiao R, Schwarz J, Throm N, Wilson AD, Benko H. MRTouch: Adding touch input to head-mounted mixed reality. IEEE Transactions on Visualization and Computer Graphics. 2018;**24**(4):1653-1660

[14] Kulshreshth A, Zorn C, LaViola JJ. Poster: Real-time markerless Kinect based finger tracking and hand gesture recognition for HCI. In: 2013 IEEE Symposium on 3D User Interfaces (3DUI). IEEE; 2013. pp. 187-188

[15] Gritti AP, Tarabini O, Guzzi J, Di Caro GA, Caglioti V, Gambardella LM, et al. Kinect-based people detection and tracking from small-footprint ground robots. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE; 2014. pp. 4096-4103

[16] Melax S, Keselman L, Orsten S. Dynamics based 3D skeletal hand tracking. In: Proceedings of the ACM

SIGGRAPH Symposium on Interactive 3D Graphics and Games. 2013. p. 184

[17] Xiao Y, Zhang Z, Beck A, Yuan J, Thalmann D. Human–robot interaction by understanding upper body gestures. Presence Teleoperators and Virtual Environments. 2014;**23**(2):133-154

[18] Sharma RP, Verma GK. Human computer interaction using hand gesture. Procedia Computer Science. 2015;**54**:721-727

[19] Cheng K, Ye N, Malekian R, Wang R. In-air gesture interaction: Real time hand posture recognition using passive RFID tags. IEEE Access. 2019;**7**: 94460-94472

[20] Ibraheem NA, Khan RZ, Hasan MM. Comparative study of skin color based segmentation techniques. International Journal of Applied Information Systems. 2013;**5**(10):24-38

[21] Song J, Sörös G, Pece F, Fanello SR, Izadi S, Keskin C, et al. In-air gestures around unmodified mobile devices. In: Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology. 2014. pp. 319-329

[22] Santos CCD, Samatelo JLA, Vassallo RF. Dynamic gesture recognition by using CNNs and star RGB: A temporal information condensation, 2019. arXiv preprint arXiv:1904.08505

[23] Wang X, Xia M, Cai H, Gao Y, Cattani C. Hidden-markov-models-based dynamic hand gesture recognition. Mathematical Problems in Engineering. 2012;**2012**:11. Article ID 986134. DOI: 10.1155/2012/986134

[24] Ranjan R, Dubey RK. Isolated word recognition using HMM for Maithili dialect. In: 2016 International Conference on Signal Processing and Communication (ICSC). IEEE; 2016. pp. 323-327

[25] Oberhauser R, Lecon C. Towards virtual reality immersion in software structures: Exploring augmented virtuality and speech recognition interfaces. 2018;**11**(1–2):34-44

[26] Boruah S, Basishtha S. A study on hmm based speech recognition system. In: 2013 IEEE International Conference on Computational Intelligence and Computing Research. IEEE; 2013. pp. 1-5

[27] Plouffe G, Cretu AM. Static and dynamic hand gesture recognition in depth data using dynamic time warping. IEEE Transactions on Instrumentation and Measurement. 2015;**65**(2):305-316

[28] Piumsomboon T, Dey A, Ens B, Lee G, Billinghurst M. The effects of sharing awareness cues in collaborative mixed reality. Frontiers in Robotics and AI. 2019;**6**(5):02

[29] Le Chénéchal M, Duval T, Gouranton V, Royan J, Arnaldi BV. Virtual immersive support for HelpiNg users an interaction paradigm for collaborative remote guiding in mixed reality. In: 2016 IEEE Third VR International Workshop on Collaborative Virtual Environments (3DCVE). IEEE; 2016. pp. 9-12

[30] Gao L, Bai H, Lee G, Billinghurst M. An oriented point-cloud view for MR remote collaboration. In: SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications. 2016. pp. 1-4

[31] Kolkmeier J, Harmsen E, Giesselink S, Reidsma D, Theune M, Heylen D. With a little help from a holographic friend: The OpenIMPRESS mixed reality telepresence toolkit for remote collaboration systems. In: Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology. 2018. pp. 1-11

[32] Hudák M, Sivý M. Web-based collaborative virtual environments to

support cross-platform access. In: Poster 2019 International Student Scientific Conference, Prague. 2019. pp. 178-182

[33] Synnott J, Nugent C, Jeffers P. Simulation of smart home activity datasets. Sensors. 2015;**15**:14162-14179