

Using mixed reality technologies to explore the exhibits of the Kuzbass Botanical Garden

Yuri I. Molorodov^{1,2}, Evgeny D. Chernyavtsev²

¹Federal Research Center for Information and Computational Technologies Novosibirsk, Russia

²Novosibirsk State University, Novosibirsk, Russia

Abstract

The elements of mixed reality technology are described, which allow you to get acquainted with some of the exhibits of the Kuzbass Botanical Garden, Kemerovo. Elements of mixed reality are integrated with an information system based on a relational database. The developed algorithms for matching elements of a 3D model to objects from a relational database in real time require the use of analysis and mathematical modeling methods.

Keywords

Virtual Reality) technology, mixed reality technology, information system, 3D model to objects, Kuzbass Botanical Garden.

1. Introduction

Currently, mixed reality has become relevant in many industries: aerospace, automotive, construction [1]. It has become in demand in medicine and education. The leading industry is games. However, if Virtual Reality (VR) technologies exist and have been actively used since 2012 [1], then the Mixed Reality (MR) industry began development relatively recently and is a promising area that has high innovation and investment potential [1]. MR technologies will allow you to take a new look at the methods of displaying virtual objects and interacting with them.

The purpose of this work is to develop a mixed reality technology for a 3D model of a botanical garden integrated with an information system based on a relational database. The development of an algorithm for matching elements of a 3D model to objects from a relational database in real time requires the use of analysis and mathematical modeling methods.

As an example, a 3D model of the Kuzbass Botanical Garden [2] with a variety of plants of various sizes and shapes is considered.

2. Domain analysis

2.1. Photogrammetric approach


The 3D model of the botanical garden was built on the basis of a photogrammetric approach [3].

SDM-2021: All-Russian conference, August 24–27, 2021, Novosibirsk, Russia

✉ yumo@ict.sbras.ru (Y.I. Molorodov)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

The reverse engineering method was chosen for the construction. Reverse engineering is the process of designing a digital model that describes an object and its technological properties by performing a comprehensive analysis of its structure. This process is aimed at creating a virtual 3D model based on an existing physical object for its study, duplication or improvement [3]. In order to create a 3D model of an object, the object must be measured using 3D scanning technologies (contact, laser, manual 3D scanners, etc.).

In this paper, a photogrammetric approach is considered, in which special photogrammetric 3D scanners are used. Photogrammetric scanners are divided into 2 types according to the principle of operation: active and passive.

Active photogrammetric scanners are a device with two cameras separated by a certain distance, with a bright LED that serves as a flash for uniform and correct scanning of the object's texture, as well as with a structured light source. The source can be an LCD projector that projects light in the form of a grid onto an object. Two cameras simultaneously receive images of a deformed grid projected on an object. Based on the obtained stereo pair of photographs, special software calculates the position of the object points relative to the scanning device using the main photogrammetric dependencies, and then displays a preliminary model on the computer screen. This scanning method is one of the fastest and most accurate, since instead of scanning one point at a time, such scanners receive data from the entire field of view at once [4].

Passive photogrammetric 3D scanners differ from active ones by the absence of a projector, so they register only the light reflected from the object in the visible range. In essence, they are paired digital cameras, but you can also use a conventional digital camera to build a model. To create three-dimensional models, photogrammetric methods are also used, which require more than 60% overlap between the images to fulfill the conditions for constructing the model [3].

The work on the survey of the selected territory of the botanical garden was carried out by specialists of the Kemerovo branch of the FRC ICT and provided to us for further use.

The shooting of the territory of the botanical garden with an area of 66 hectares was carried out from a height of 100 meters on the DJI Mavic 2 Pro unmanned aerial vehicle with a Hasselblad L1D-20c camera with a resolution of 20 megapixels. The resolution of the ortho-photoplane was 2.49 cm/px (45280×45860 px).

The territory was divided into 3 parts with dimensions of 17 hectares, 27 hectares and 22 hectares and 3 polygons were filmed. This is due to the fact that the copter from one battery can fly for 25–28 minutes (there were 3 batteries). The average flight speed is 8 m/s, the camera angle is 90° , the percentage of overlap of images is 70%. As a result, 973 images were obtained in JPEG format with a set of metadata in EXIF format — camera parameters, as well as geographical coordinates and altitude above sea level.

The flight plan was compiled in the Pix4Dcapture program (installed on the DJI Smart Controller). The received photos were processed using the following software: Pix4dapper and Agisoft Metashape.

A point cloud (point model) is a set of points obtained as a result of 3D scanning of a real-world object and representing the surface of this object in a three-dimensional coordinate system. Cloud points are usually represented by XYZ coordinates, which are simply written in a file.

Point clouds provide fast visualization of a real-world object. At the same time, they are successfully used for measuring and controlling objects, 3D printing, visual visualization of hard-to-reach places or large extended objects, creating three-dimensional and mathematical

models, pattern recognition, automated analysis, reconstruction and operation, and are also the basis for reverse engineering of real-world objects.

2.2. Mixed reality

In the modern world, there are several types of reality: virtual, augmented and mixed [5].

Mixed reality is called hybrid and is a model of world perception in which the real and virtual worlds are combined. This model is also called Mixed Reality or MR for short.

Back in the mid-1990s, it was possible to establish that in the strict sense of the word, all modern reality can be considered mixed, since it is located between two “extreme points” — the completely real, physical world and the completely virtual, artificially created world of augmented or virtual reality.

The original mixed reality technologies were developed back in the 1970s and 1980s, although at that time their modern understanding had not yet been achieved. As is usually the case, first of all, MR was developed for military and training purposes, and only then similar technologies were transferred to the commercial sphere and to the entertainment industry.

What are the differences between mixed reality and virtual and augmented reality? Usually, terms such as MR, VR, AR are confused with each other and used as synonyms, although in fact there are a number of characteristic features by which they differ.

- The real world is an objective reality, not supplemented by artificial technologies.
- Virtual reality (VR) is a subjective, unreal world created with the help of screens, holograms and other artificial means.
- Augmented Reality (AR) — a digitized real world with hints, holograms, and other objects superimposed on top of it. This is a virtual world that is built on the basis of the real one and obeys it in everything.
- Mixed reality (MR) differs from AR in that virtual objects are able to influence the real world, and not just obey it.

With the help of MR, it is possible to conduct simulation-based training, military training without increased risks. Also, the development of applications in this area is aimed at creating an interactive environment with the full inclusion of virtual objects in reality, using such objects for commercial, developmental, and entertainment purposes [6].

So we set ourselves the task of creating a mixed reality application for a 3D model of a botanical garden for educational purposes.

Next, information will be provided about the software packages and equipment required to create MR applications.

The following components must be installed:

- Windows 10;
- Visual Studio 2019;
- SDK package for Windows 10 (10.0.18362.0);
- HoloLens emulator of the first or second generation (you can use the glasses themselves, but for lack of it, you can limit yourself to an emulator).

All components can be downloaded from the official Microsoft website [7].

As for the development tools, you need to download and import the Mixed Reality Toolkit (MRTK). MRTK is an open source cross-platform mixed reality application development kit. MRTK provides a cross-platform input system, basic components and common building blocks for spatial interaction. The toolkit is designed to accelerate the development of applications for Microsoft HoloLens, Windows Mixed Reality immersive (VR) headsets and the OpenVR platform.

After installing the MRTK, you need to configure the configuration in Unity to use mixed reality tools.

After setting up, we can create a scene in Unity, build the application, and use Visual Studio to connect HoloLens to the computer, deploy our application.

All additional development features are described on the Microsoft website in the Unity Development section [8]. Capabilities are understood as: placing objects in a scene, creating dynamic content using solvers, creating user interfaces, tracking the gaze, using voice commands, etc. A detailed description of modeling interaction with 3D objects is given in [9].

2.3. Game engine

The most popular game engines today are Unreal Engine (UE4) and Unity 3D (Unity). Each of the engines has its own strengths for different tasks. Unity is suitable for beginners and amateurs, while UE4 is strictly for professional developers. UE4 is more flexible in terms of animation and three-dimensional graphics, but Unity shows itself better in MR development. We chose Unity [10] to work with.

Figure 1 shows the interface of the Unity program with an open empty project.

The program interface is divided into 4 areas. The “1” area (hereinafter referred to as the “object panel”) displays objects that are located in the “3” area (hereinafter referred to as the “scene”). Initially, in an empty project, a camera and a directional light are installed on the stage so that we can see what is happening and distinguish objects using the shadows they cast. In

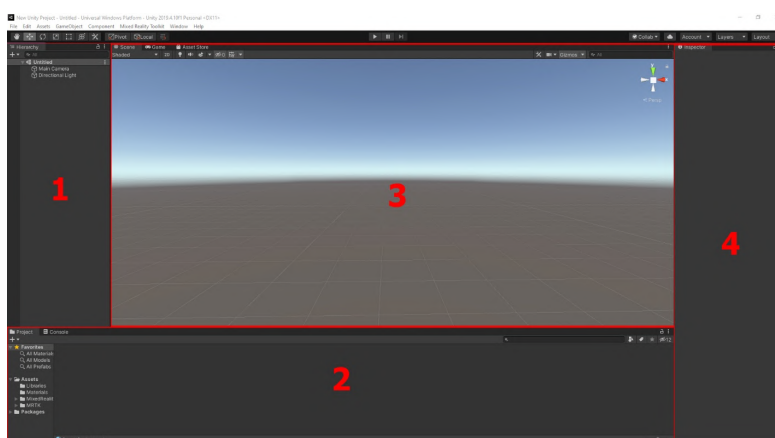


Figure 1: Unity interface.

the “2” area (hereinafter referred to as the “component panel”) there are materials necessary for the project: textures, 3D models, scripts, etc. They are needed to define the overall concept of the application. On the stage, we can arrange objects from the objects panel as we want. We can fly in it, controlling the mouse and keyboard keys, select objects, move and transform them. You can select the method of manipulating an object in the upper-left corner of the window (above the objects panel). And finally, in the “4” area (hereinafter referred to as the “properties panel”), when you click on any object or material from the “1” or “2” area, their properties are displayed that we can change, which makes our application unique because of its flexibility [11].

Programming scripts for interactive interaction in Unity takes place in the C# language. When creating a script, a template is initially installed in the file, which has the connection of the necessary libraries and two procedures called Start and Update. What is written in the Start procedure block is executed when the program is started, and what is written in Update is executed every frame. There is special documentation on programming for Unity [12].

2.4. Creating a scene in the game engine

In Unity, we create a new project. We import a ready-made 3D model of the botanical garden from the components panel into our project. We transfer it to the objects panel and the model appears on the scene relative to the beginning of the local coordinate system.

In the objects panel, we create an empty object called FirstPersonController and a camera called Camera. In the objects panel, we fix the camera on the observer. We can place the scene in any place of the observer, and the camera at the level of his head. To control the observer, you need to add the Character Controller component in the properties panel. To configure the movement of the observer, we add a control program-script to the component panel. We'll link her to the camera, now she can move with him. To enable the camera to rotate using the mouse cursor, we wrote and linked a script to the object.

Interaction (collision) of the observer with the exhibits the botanical garden has developed a collision program (an approximately created polygonal network). To ensure a collision, a complex collision was programmed (a network generated exactly according to the shape of the model). Any other collision will not be able to give the effect of full immersion in the world of the botanical garden, since it is necessary to move around the territory of the botanical garden in accordance with its relief. The application is launched by clicking the Start button at the top of the window. This allows the observer to move and rotate the “head”.

To accurately determine the position of plants on the scene, the coordinates of the observer are fixed relative to the origin of the coordinate system. This is provided by the corresponding part of the script code.

To determine the coordinates of the plant that the observer is looking at, a white circle is set in the center of the screen. Using the script, we can determine the coordinates of the object that this circle points to.

To display the coordinates of an object on the scene, two text fields have been added in the upper-left corner of the screen: one for displaying the location of the observer, and the second for displaying the coordinates of his “view”. Thus, during the operation of the application, when the observer moves, its coordinates are updated each time and translated into the text field, and when you click on an object on the scene (according to the model of the botanical garden),

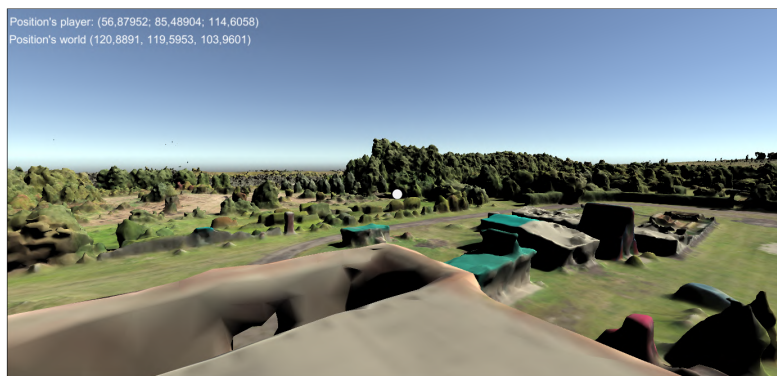


Figure 2: Getting the coordinates of the character and objects on the stage.

the coordinates corresponding to the location of this object relative to the origin are displayed (Figure 2).

Every time the application is launched, the model of the botanical garden is completely redrawn.

2.5. Relation of the database to the botanical garden model

The model of the botanical garden is integrated into the information system using a relational database. A shape file is generated from this database (a vector format for storing objects described by geometry and related attributes). The territory of the botanical garden is geometrically divided into sections with the attributes of the description and the corresponding coordinates (Figure 3).

The result of the work of the program fragment to determine the coordinates of infrastructure objects is shown in Figure 4.

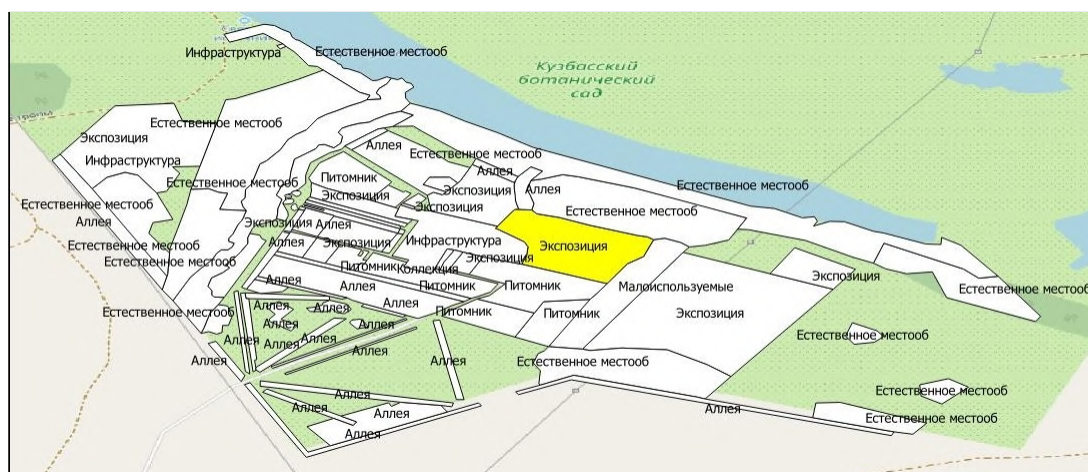


Figure 3: The shape file of the botanical garden.



Figure 4: Coordinates of the selected DB objects.

The shape file can be supplemented with objects, dividing the territory into smaller sections for more variety. But the task of the project is the ability to display information about individual plant species located in the botanical garden. Therefore, in order to narrow down the problem, we used the source of storing a huge database with a detailed description of each plant and the corresponding world coordinates (longitude and latitude) [13].

To navigate through the territory of the botanical garden, it is necessary to determine the coordinates and extract information about infrastructure objects. We will take the information necessary for the work from the database.

The database consists of one table containing more than 2000 records about plant species (Figure 5).

The table has the following fields: unique identifier, author's name, comment, date, description and location. The ID and location fields have a numeric data type, and the other fields have a text data type.

The database is accessed by a separate script, which is added as a component to any element

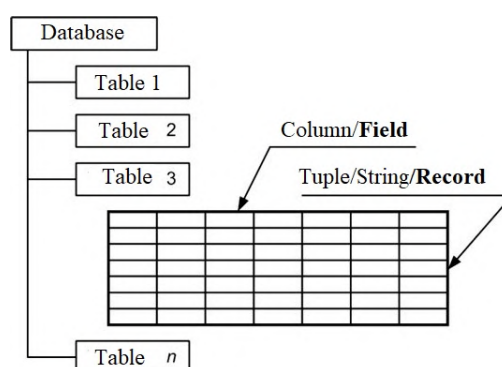


Figure 5: Relational database structure.

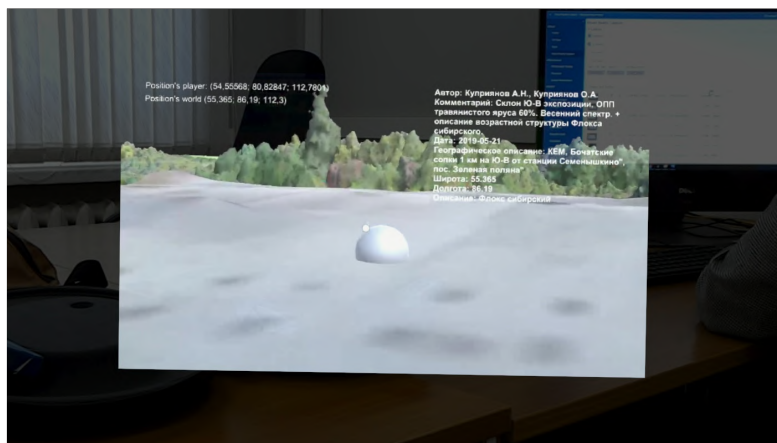


Figure 6: The application works with mixed reality glasses.

on the scene. It provides a connection to a local server, lists are formed for each field, for the subsequent selection of information from the table. When you click on an object on the model of the botanical garden, the coordinates are transmitted to the server in the form of an SQL query [14]. As a result of executing the query, unnecessary records that do not correspond to the selected location are cut off, and a single table entry with detailed information about the plant species is displayed in the form of a graphic image on the screen (Figure 6).

2.6. Uploading to Mixed Reality glasses

First of all, we will make sure that there is a set of application development tools for Microsoft HoloLens, Windows Mixed Reality immersive (VR) headsets and the OpenVR platform. If there are no virtual reality glasses, then we configure work with the HoloLens emulator of the first or second generation. The necessary components can be downloaded from the official Microsoft website [7].

Next, we turn on the developer mode both on the computer and on the HoloLens glasses [15] and perform the procedure for configuring the application build configuration for the selected platform [16].

Upon completion of the work, a file supported by the Visual Studio 2019 program will be created in the previously specified folder. We open it. Select the x86 configuration for the application, select Release as the result of debugging, select the Remote Machine deployment method (remote computer).

The Connection Settings dialog box will automatically appear. Enter the IP address of your device in the Address or Computer Name field. This IP address can be found in HoloLens by selecting Settings → Network & Internet → Advanced Options. We set the Universal authentication mode (unencrypted protocol). Select Debug, then Start Debugging to deploy the application and start debugging. When you first deploy the application from your computer to HoloLens, you will be prompted to enter a PIN code. We enter the PIN code, after which we see how after a while our application appears in the glasses, which we can launch and do the same as in the Unity game engine [17].

3. Conclusion

In the Unity game engine, the scene was simulated, the database was connected to the project and loaded into HoloLens mixed reality glasses using a ready-made 3D model of the botanical garden scanned using an unmanned aerial vehicle. The process of modeling mixed reality technology from creating a 3D model to implementing it in mixed reality glasses has been developed.

This application will be useful both for educational purposes and for serious study of nature in general. Applications of this format can be created in any field of activity, and this is thanks to the technologies provided by mixed reality. The development of technology will allow us to explore the world in various aspects of its diversity.

References

- [1] Relevance and prospects of mixed reality. Available at: <http://samag.ru/archive/article/3882>.
- [2] Kuzbass Botanical Garden. Available at: <http://kuzbs.ru>.
- [3] Three-dimensional photogrammetry. Available at: <https://sapr.ru/article/25136>.
- [4] Photogrammetric method for creating a 3D model. Available at: <https://innotech.ua/ru/news/fotogrammetriya-metod-sozdaniya-detalizirovannoy-3d-modeli-iz-fotografii-61122>.
- [5] Types of realities. Available at: <https://www.it.ua/ru/knowledge-base/technology-innovation/dopolnennaja-virtualnaja-i-prochie-realnost>.
- [6] Mixed reality technology. Available at: https://funreality.ru/technology/mixed_reality.
- [7] Mixed reality Development Tools. Available at: <https://docs.microsoft.com/ru-ru/windows/mixed-reality/develop/install-the-tools?tabs=unity>.
- [8] Additional features of mixed reality development. Available at: <https://docs.microsoft.com/ru-ru/windows/mixed-reality/develop/unity/tutorials/mr-learning-base-01>.
- [9] Modeling of interaction with 3D objects. Available at: <https://docs.microsoft.com/ru-ru/windows/mixed-reality/develop/unity/mrtoolkit-101>.
- [10] Comparison of UE4 and Unity3D game engines. Available at: <https://dtf.ru/gamedev/7227-orel-ili-reshka-sravnenie-unity-i-unreal-engine>.
- [11] The Unity3D game engine. Available at: <https://unity.com/ru>.
- [12] Unity3D documentation. Available at: <https://docs.unity3d.com/Manual/index.html>.
- [13] Open access to data on biodiversity. Available at: <https://www.gbif.org/ru>.
- [14] Basic SQL commands. Available at: <https://tproger.ru/translations/sql-recap>.
- [15] Setting up the connection of the computer with mixed reality glasses. Available at: <https://docs.microsoft.com/ru-ru/windows/mixed-reality/develop/platform-capabilities-and-apis/using-visual-studio>.
- [16] Building the application in Unity. Available at: <https://unity3d.com/ru/partners/microsoft/porting-guides>.
- [17] Deploying the application in the Visual Studio environment. Available at: <https://docs.microsoft.com/ru-ru/windows/mixed-reality/develop/platform-capabilities-and-apis/using-visual-studio>.