

METHOD, SYSTEM AND ELECTRICITY DEVICE FOR DISPLAYING MIXED REALITY (MR) SCENES

TECHNICAL FIELD

[0001] This application relates to mixed reality technology, and more particularly to a method, system and electricity device for displaying MR scenes.

BACKGROUND

[0002] MR technology is an advancement of virtual reality (VR) technology. By presenting virtual scene information within real-world environments, MR technology creates an interactive feedback loop between the real world, the virtual world and the user, enhancing the realism of the user experience.

[0003] However, due to the current limitations in MR interaction, virtual content and real-world scenes cannot achieve perfect alignment, resulting in a lack of synchronization between objects of the real world and their counterparts in virtual world, which ultimately impacts the user experience.

SUMMARY

[0004] 1. Technical issues

[0005] To address the shortcomings of the prior art, the present disclosure provides a method, system, and electronic device for displaying MR scenes, solving the technical problem of incomplete interaction in mixed reality, where virtual content and real-world scenes cannot achieve seamless fusion.

[0006] 2. Technical solution

[0007] To achieve the above objects, the present disclosure provides the following technical solution.

[0008] Firstly, the present application provides a method for displaying MR scenes, comprising:

(S1) obtaining a point cloud data of a real-world scene, wherein a plurality of physical markers are provided within the real-world scene and have a three-dimensional

structure with an uneven surface;

(S2) identifying each of the plurality of physical markers through a classification algorithm; and obtaining a pose of each of the plurality of physical markers and a position of each of the plurality of physical markers, as well as a relative relationship between the plurality of physical markers, through point cloud registration; and

(S3) adjusting a size of a virtual content, a pose of the virtual content and a position of the virtual content based on the pose of each of the plurality of physical markers and the position of each of the plurality of physical markers as well as the relative relationship between the plurality of physical markers, and aligning a virtual marker within the virtual content with the plurality of physical markers through point cloud registration, so as to achieve an alignment of the virtual content on a surface of a physical object.

[0009] In an embodiment, the real-world scene has at least two physical markers, and a positional relationship between at least two physical markers among the plurality of physical markers satisfies the following condition:

a mounting surface of a first physical marker is on a first plane, and a mounting surface of a second physical marker is on a second plane; the first plane is perpendicular to the second plane;

the first plane intersects the second plane along a line; and

the closest point on the line to a center of the mounting surface of the first physical marker is the same as the closest point on the line to a center of the mounting surface of the second physical marker.

[0010] In an embodiment, the plurality of physical markers are formed by stacking a plurality of cubes; so as to form the three-dimensional structure with the uneven surface.

[0011] In an embodiment, the uneven surface comprises an image with a texture.

[0012] In an embodiment, the step of identifying each of the plurality of physical markers by the classification algorithm comprises:

extracting a geometric feature and a surface feature from the point cloud data;

combining the geometric feature and the surface feature into feature vectors to characterize each of the plurality of physical markers; and

classifying the feature vectors by a machine learning algorithm to distinguish the plurality of physical markers.

[0013] In an embodiment, the step of adjusting the size of the virtual content, the pose of the virtual content and the position of the virtual content based on the pose of each of the plurality of physical markers and the position of each of the plurality of physical markers as well as the relative relationship between the plurality of physical markers comprises:

calculating an overall change rate between a current frame and a previous frame of the real-world scene;

dynamically adjusting a feedback frequency based on the overall change rate, wherein the feedback frequency refers to a rate at which an alignment between the virtual content and the real-world scene is updated and adjusted according to the overall change rate;

calculating an alignment error between the virtual content and the real-world scene;

determining an optimization algorithm based on the alignment error and a scene complexity; and

adjusting the size of the virtual content, the position of the virtual content and the pose of the virtual content according to the optimization algorithm.

[0014] In an embodiment, the step of calculating the overall change rate between the current frame and the previous frame of the real-world scene comprises:

calculating a rate of change in the position of a physical marker among the plurality of the physical markers in the real-world scene between the current frame and the previous frame to obtain a position change rate, and calculating a rate of change in the pose of the physical marker in the real-world scene between the current frame and the previous frame to obtain a pose change rate; and

obtaining the overall change rate based on the position change rate and the pose change rate through the following formula:

$$R_c = \alpha \cdot R_p + \beta \cdot R_s;$$

wherein R_c represents the overall change rate, R_p represents the position change

rate, R_s represents the pose change rate, α and β are weighting coefficients.

[0015] In an embodiment, the step of dynamically adjusting the feedback frequency based on the overall change rate comprises:

setting an initial feedback frequency, and dynamically adjusting the feedback frequency based on the overall change rate through the following formula:

$$F = F_b + (1 + k \times R_c)$$

wherein, F_b represents the initial feedback frequency, F represents the feedback frequency, k is an adjustment coefficient, and R_c represents the overall change rate.

Secondly, the step of determining the optimization algorithm based on the alignment error and the scene complexity comprises:

setting a threshold for a positional error within the alignment error, a threshold for the angular error within the alignment error and a threshold for the scene complexity; and

determining whether the positional error is less than the threshold for the positional error, the angular error is less than the threshold for the angular error, and the scene complexity is less than the threshold for the scene complexity, if the above condition are all met, determining a local optimization algorithm, otherwise, determining a global optimization algorithm.

[0016] Secondly, the present application provides a system for displaying MR scenes, comprising:

- a point cloud acquisition module;
- a marker identification module; and
- a registration and fusion module;

wherein the point cloud acquisition module is configured to obtain a point cloud data of a real-world scene, wherein a plurality of physical markers are provided within the real-world scene, the plurality of physical markers has a three-dimensional structure with an uneven surface;

the marker identification module is configured to identify each of the plurality of physical markers through a classification algorithm; and to obtain a pose of the plurality

of physical markers and a position of the plurality of physical markers, as well as a relative relationship between the plurality of physical markers, through point cloud registration; and

the registration and fusion module is configured to adjust a size of a virtual content, an angle of the virtual content and a position of the virtual content, and to align a virtual marker within the virtual content with the plurality of physical markers through point cloud registration, so as to achieve an alignment of the virtual content on a surface of a physical object.

[0017] Thirdly, the present application provides an electronic device, comprising:

- a processor;
- a memory; and
- a program;

wherein the program is stored in the memory and configured to be executed by the processor, the program comprises the method for displaying MR scenes.

[0018] 3. Benefits

[0019] The present disclosure provides a method, system, and electronic device for displaying MR scenes. Compared to existing technology, it offers the following benefits.

[0020] The present application adopts physical markers with the unique structure with uneven surfaces, providing rich geometric features such as corner points, edges, and curvature changes. These features are more easily identified and matched by point cloud data. By leveraging these prominent geometric features for feature extraction and characterization, the initial alignment accuracy of point cloud registration is improved. This enables the automatic identification and feature localization of real-world scenes, allowing virtual content to be accurately aligned and fit on their surfaces.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] In order to more clearly illustrate the technical solutions in the embodiments of the present disclosure or in the prior art, the accompanying drawings needed in the description of the embodiments or the prior art will be briefly described below. Obviously, presented in the accompanying drawings are only some embodiments of the

present disclosure. Other embodiments may also be obtained by those skilled in the art according to these accompanying drawings without making creative effort.

[0022] Fig. 1 is a flowchart of a method for displaying MR scenes according to Embodiment 1 of the present disclosure;

[0023] Fig. 2 schematically shows a real-world scene (a picture frame as an example) with three physical markers according to Embodiment 2 of the present disclosure;

[0024] Fig. 3 schematically shows the positional relationship between each physical marker and the real-world scene, when there are two physical markers in the real-world scene, according to Embodiment 2 of the present disclosure; and

[0025] Fig. 4 schematically shows the positional relationship between each physical marker and the real-world scene, when there are three physical markers in the real-world scene, according to Embodiment 2 of the present disclosure.

DETAILED DESCRIPTION OF EMBODIMENTS

[0026] To make the objects, technical solutions and advantages of the embodiments of this application clearer, the technical solutions of the embodiments of this application will be clearly and completely described. It is obvious that described below are some embodiments instead of all embodiments of this application. Based on the embodiments in this application, all other embodiments obtained by those skilled in the art without making creative effort shall fall within the scope of this present application.

[0027] The present disclosure provides a method, system, and electronic device for displaying MR scenes, solving the technical problem of incomplete interaction in mixed reality, where the seamless integration between virtual content and real-world scenes cannot be achieved. This enables the automatic identification and feature localization of the real-world scenes, allowing the virtual content to be accurately aligned and fit on the surfaces of the real-world scenes.

[0028] The technical solution in this embodiment of the present application aims to solve the above technical problems, and the overall idea is as follows.

[0029] Although MR headsets and other devices have already emerged, there are still technical challenges in achieving automatic identification of real-world scenes and the

natural display of virtual content in MR scenes. Traditional methods often require manual adjustment of the position and size of virtual content, lacking automation and intelligence, which negatively impacts user experience and display quality.

[0030] With the development of MR technology, how to seamlessly integrate virtual content with real-world scenes in MR environments has become a critical issue. The current technology faces the following challenges: how to achieve automatic identification and feature localization of real-world scenes to accurately align virtual content on their surfaces.

[0031] This present application adopts MR headset devices to identify the real-world scenes with specific features and project 3D content, videos, or 2D photos from the MR environment onto the surfaces of the real-world scenes. When users focus on the real-world scene for an extended period, 3D interactive elements reproduce, allowing users to interact by clicking to switch 3D content and perform other functions. The goal of the present application is to provide an innovative display method that enhances the appeal and immersion of the displayed content while also improving user engagement and interactive experience.

[0032] For a better understanding of the present disclosure, the technical solutions of the embodiments of this application will be clearly and completely described below in conjunction with the accompanying drawings.

[0033] Embodiment 1

[0034] As shown in Fig. 1, this embodiment provides a method for displaying MR scenes, and the method includes the following steps.

[0035] (S1) A point cloud data of a real-world scene is obtained. A plurality of physical markers are provided within the real-world scene and have a three-dimensional structure with an uneven surface.

[0036] (S2) Each of the plurality of physical markers is identified through a classification algorithm. A pose of each of the plurality of physical markers and a position of each of the plurality of physical markers, as well as a relative relationship between the plurality of physical markers, are obtained through point cloud registration.

[0037] (S3) A size of a virtual content, a pose of the virtual content and a position of

the virtual content are adjusted based on the pose of each of the plurality of physical markers and the position of each of the plurality of physical markers as well as the relative relationship between the plurality of physical markers. A virtual marker within the virtual content is aligned with the plurality of physical markers through point cloud registration, so as to achieve an alignment of the virtual content on a surface of a physical object.

[0038] The embodiment adopts physical markers with a unique structure with uneven surfaces, providing rich geometric features such as corner points, edges, and curvature changes. These features are more easily identified and matched by point cloud data. By leveraging these prominent geometric features for feature extraction and characterization, the initial alignment accuracy of point cloud registration is improved. This enables the automatic identification and feature localization of real-world scenes, allowing virtual content to be accurately aligned and fit on their surfaces.

[0039] Embodiment 2

[0040] (S1) A point cloud data of a real-world scene is obtained. A plurality of physical markers are provided within the real-world scene and have a three-dimensional structure with an uneven surface. Step (S1) includes the following sub-steps.

[0041] As shown in Fig. 2, the plurality of physical markers are formed by stacking a plurality of cubes; so as to form the three-dimensional structure with the uneven surface. The surface of these physical markers is covered with images featuring distinct textures. In this embodiment, three physical markers with different structures are provided. It should be noted that, for an easier understanding of the relationships between the physical markers, a picture frame is used as an example of the real-world scene in this embodiment.

[0042] The mounting surfaces (fixed surfaces) of the three physical markers conform to the following rules.

[0043] The mounting surfaces of physical marker A and physical marker B are on the same plane, plane_A; the mounting surface of physical marker C is on plane_B, plane_B is perpendicular to plane_A. Additionally, plane_C formed by the lines connecting the three physical markers, is perpendicular to plane_A.

[0044] In specific implementations, for easier installation, two physical markers may be installed (i.e., removing physical marker B from Fig. 2), and the mounting surfaces (i.e., fixed surfaces) of the two physical markers conform to the following rules.

[0045] The mounting surface of physical marker A is on plane_a, and the mounting surface of physical marker C is on plane_b. Plane_b is perpendicular to plane_a.

[0046] Plane_a intersects plane_b along line_a.

[0047] The closest point on line_a to the center of the mounting surface of physical marker A is the same as the closest point on line_a to the center of the mounting surface of physical marker B.

[0048] From the above description, it can be seen that in the case of three physical markers, two of the physical markers also meet the rules for the two physical markers. Thus, in the embodiment of the present application, the real-world scene includes at least two physical markers, and when there are multiple physical markers, at least two physical markers satisfy the above positional relationship between physical marker A and physical marker C.

[0049] (S2) Each of the plurality of physical markers is identified through a classification algorithm; and a pose of each of the plurality of physical markers and a position of each of the plurality of physical markers, as well as a relative relationship between the plurality of physical markers are obtained, through point cloud registration. Step (S2) includes the following sub-steps.

[0050] The process of identifying each physical marker using a classification algorithm is as follows.

[0051] a. Geometric features and surface features in the point cloud data can be used to characterize each physical marker. This can involve calculating geometric features such as normal vectors, curvature, and curvature changes around each point, as well as surface features such as point color and texture.

[0052] b. The extracted features are combined into a feature vector that characterizes each physical marker. Dimensionality reduction techniques like PCA (Principal Component Analysis) or others can be used to reduce the dimensionality of the feature vector, making it more efficient to represent the structure's characteristics.

[0053] c. A machine learning algorithm can be used to classify the feature vectors and distinguish between the physical markers (in the embodiment of the present application, the physical markers are three). A classification algorithm suitable for the data characteristics and scale can be selected, such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree, or Random Forest.

[0054] It should be noted that in the embodiment of the present application, the point cloud data is collected using the following method. The target area is scanned by laser intensity detection and ranging (LiDAR) or a camera combined with a depth sensor to obtain point cloud data containing physical markers A, B, and C. Before using the point cloud data, the collected data must be preprocessed, including denoising, downsampling, and filtering, to ensure data quality and processing efficiency.

[0055] It should be noted that during the stage of training, labeled point cloud data must be provided to allow the algorithm to learn how to classify different feature vectors into three distinct categories. Once training is completed, the classification algorithm can be applied to new point cloud data to identify and distinguish the three physical markers.

[0056] The step of obtaining a pose of each of the plurality of physical markers and a position of each of the plurality of physical markers, as well as a relative relationship between the plurality of physical markers, through point cloud registration, includes the following sub-steps.

[0057] Point cloud registration includes both initial registration and fine registration.

[0058] The process of the initial registration involves using a feature matching algorithm (such as RANdom SAMple Consensus (RANSAC)) to perform a rough alignment of the initial feature points, resulting in a preliminary alignment outcome. By estimating the relative transformation matrix, a rough estimate of the pose and position of the plurality of physical markers are obtained.

[0059] The fine registration process involves using the Iterative Closest Point (ICP) algorithm for fine registration to enhance the accuracy of the alignment. During the iterative process, the distance error between the target point cloud and the source point cloud is minimized to optimize the transformation matrix.

[0060] Pose and Position Calculation:

[0061] The pose and position of physical markers A, B, and C are calculated by using the transformation matrix obtained from the fine registration. Based on the registration results, the relative positional relationships between the physical markers are determined.

[0062] The calculation of relative relationships specifically involves determining the positions of the physical markers in space by calculating the center point of each of the physical markers. Based on the pose of each physical marker, the relative angles and distances between the physical markers are calculated. The relative position and pose information is then used to construct a relative relationship matrix, which is used for further alignment and display of virtual content.

[0063] (S3) A size of a virtual content, a pose of the virtual content and a position of the virtual content are adjusted based on the pose of each of the plurality of physical markers and the position of each of the plurality of physical markers as well as the relative relationship between the plurality of physical markers. A virtual marker within the virtual content is aligned with the plurality of physical markers through point cloud registration, so as to achieve an alignment of the virtual content on a surface of a physical object.

[0064] The geometric features of the virtual markers are matched with the physical markers. The geometric dimensions and shapes of the virtual markers are highly consistent with those of the physical markers, ensuring that the geometric accuracy between the virtual markers and the physical markers is maintained.

[0065] Based on the feature points and positional information of the real-world scene, the size, pose, and position of the virtual content are automatically adjusted according to the length, width, and height of the real-world scene, achieving precise alignment of the virtual content on the surface of physical objects.

[0066] In the embodiments of the present application, the pose, position and size of the virtual content are adjusted through an adaptive real-time feedback mechanism.

[0067] Traditional feedback mechanisms have issues with lag and fixed frequency when handling dynamically changing MR scenes. To address this, an adaptive real-time

feedback mechanism is proposed, which dynamically adjusts the feedback frequency and optimization strategy based on changes in sensor data, ensuring continuous alignment of the virtual content with the real-world scene.

[0068] By selecting appropriate optimization algorithms and dynamically adjusting optimization parameters, the alignment between the virtual content and the real-world scene can be significantly improved. When the error is large and the scene is complex, a global optimization algorithm is employed to find the overall optimal solution; when the error is small and the scene is simple, a local optimization algorithm is used for rapid convergence and improving efficiency. Dynamically adjusting the parameters of the optimization algorithm further enhances optimization effectiveness and system adaptability.

[0069] The specific process for adjusting the post and size of virtual content using the adaptive real-time feedback mechanism is as follows.

[0070] (S301) The point cloud data and a pose data of the real-world scene are obtained in real time; and an overall change rate between a current frame and a previous frame of the real-world scene is calculated. Step (S301) includes the following sub-steps.

[0071] In this embodiment, a rate of change in the position of a physical marker among the plurality of the physical markers in the real-world scene between the current frame and the previous frame is calculated to obtain a position change rate, and a rate of change in the pose of the physical marker in the real-world scene is calculated between the current frame and the previous frame to obtain a pose change rate. The overall change rate is obtained based on the position change rate and the pose change rate.

[0072] Assume that the position information and pose information at a certain moment are as follows:

[0073] $P_c=(x_1, y_1, z_1)$; P_c represents the position of the current frame.

[0074] $P_p=(x_0, y_0, z_0)$; P_p represents the position of the previous frame.

[0075] $Q_c=(w_1, x_1, y_1, z_1)$; Q_c represents the pose of the current frame (quaternion).

[0076] $Q_p = (w_0, x_0, y_0, z_0)$; Q_p represents the pose of the previous frame (quaternion).

[0077] The time interval is calculated as the following formula:

[0078] $\Delta t = t_1 - t_0$; Δt represents the time interval.

[0079] The position change rate is calculated through the following formula:

[0080] $R_p = \frac{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2}}{\Delta t}$; R_p represents the position change rate.

[0081] $\Delta Q = Q_c \cdot Q_p^{-1}$;

[0082] After quaternion multiplication and inversion, the pose change rate is calculated through the following formula:

$$\Delta Q = (w\Delta, x\Delta, y\Delta, z\Delta)$$

$$R_s = \frac{2 \cdot \cos^{-1}(w\Delta)}{\Delta t}$$

[0083] R_s represents the pose change rate.

[0084] The overall change rate is calculated through the following formula:

[0085] $R_c = \alpha \cdot R_p + \beta \cdot R_s$

[0086] R_c represents the overall change rate; and α and β are weighting coefficients.

[0087] In the specific implementation process, the change rate can also be calculated using other methods, such as changes in the MR headset's position and orientation.

[0088] (S302) A feedback frequency is dynamically adjusted based on the overall change rate. The feedback frequency refers to a rate at which an alignment between the virtual content and the real-world scene is updated and adjusted according to the overall change rate. Step (S302) includes the following sub-steps.

[0089] The feedback frequency refers to how often the alignment between the virtual content and the real-world scene is updated and adjusted based on changes in sensor data. A high feedback frequency allows for a quick response to changes in the real-world scene but increases the computational load; a low feedback frequency reduces the computational burden but may result in lag.

[0090] The feedback frequency is dynamically adjusted based on the overall change

rate. If the overall change rate is high, the feedback frequency is increased; if the overall change rate is low, the feedback frequency is decreased. An initial feedback frequency, such as 30 times per second, is set. The feedback frequency is dynamically adjusted based on the overall change rate: if the change rate increases, the feedback frequency is raised to ensure a quick response. If the change rate decreases, the feedback frequency is lowered to conserve computational resources. In the implementation of this application, the feedback frequency is dynamically adjusted using an adjustment coefficient.

[0091] $F = F_b + (1 + k \times R_c)$

[0092] In the above formula, k is an adjustment coefficient that can be adjusted based on the actual situation; R_c represents the overall change rate; F represents the feedback frequency; F_b represents the initial feedback frequency.

[0093] (S303) An alignment error between the virtual content and the real-world scene is calculated. Step (S303) includes the following sub-steps.

[0094] The alignment error includes positional error and angular error.

[0095] The positional error is the average of the distances between the positions of the virtual content plane and the real-world scene plane over n frames.

[0096] The angular error is the average of the angles between the normal vectors of the virtual content plane and the real-world scene plane over n frames.

[0097] (S304) An optimization algorithm is determined based on the alignment error and a scene complexity. Step (S304) includes the following sub-steps.

[0098] In the specific implementation process, there are two methods for evaluating scene complexity.

[0099] The first method for evaluating scene complexity is as follows. Scene complexity is assessed based on the number of feature points, with scene complexity being directly proportional to the number of feature points. Assuming the scene complexity is 1 when the number of feature points is 1000, and the following formula is obtained.

[0100] $SC = NF/1000$

[0101] In the above formula, NF represents the number of feature points and SC represents the scene complexity.

[0102] The second method for evaluating scene complexity is as follows. Scene complexity (SC) is evaluated by taking into account the number of feature points (NF), the mean curvature (MC), and the texture complexity (TC) as follows.

[0103] $SC = NF/1000 + MC \times a + TC \times b$

[0104] In the above formula, a and b are weighting coefficients. In this embodiment, $a = 0.5$; $b = 2$.

[0105] In this embodiment, the threshold for the positional error is set to 0.1, the threshold for the angular error is set to 15, and the threshold for the scene complexity is set to 15.

[0106] If all the above three indicators are below their respective thresholds, a local optimization algorithm is selected; otherwise, a global optimization algorithm is chosen.

[0107] Through validation and testing, it is generally found that a global optimization algorithm is chosen for scenarios such as industrial equipment maintenance, while a local algorithm is used for simpler scenarios such as virtual tours of conference rooms or museum exhibits.

[0108] In the specific implementation process, optimization algorithm parameters can be dynamically adjusted to enhance optimization effectiveness. For instance, in genetic algorithms, population size, crossover probability, and mutation probability can be adjusted. In gradient descent algorithms, the learning rate and number of iterations can be modified.

[0109] (S305) The position of the virtual content and the pose of the virtual content are adjusted according to the optimization algorithm to ensure accurate alignment between the real-world scene and the virtual content.

[0110] The problem of adjusting the position and posture is defined as an optimization problem. The goal is to minimize the error between the virtual content and the real-world scene, where the error consists of positional error and post error. This alignment primarily relies on optimization algorithms that continuously adjust the position and post of the virtual content to ensure the highest degree of matching between the virtual

content and the real objects in the real-world scene.

[0111] In the embodiment of the present application, the positional error is defined as the Euclidean distance between the corresponding points of the virtual content and the real object.

[0112] The post error is defined as the angle between the normal vectors of the corresponding points of the virtual content and the real object.

[0113] In the embodiment of the present application, the objective function (EEE) is constructed to represent the total error. The objective function is represented by the following formula.

$$\textbf{[0114]} \quad E = \sum_{i=1}^N (\|V_i - P_i\|^2 + z(1 - \cos(\varphi_i)))$$

[0115] In the above formula, V_i represents the position of the i -th point of the virtual content, P_i represents the position of the i -th point of the real object, φ_i is the angle between the normal vectors of the virtual content point and the real object point, z is weight parameter that balances the positional error and posture error, and N is the total number of points. The total number of points of the points of the real object is equal to the total number of the points of the virtual content.

[0116] In the embodiment of the present application, various optimization algorithms can be used to minimize the objective function (EEE), such as global optimization algorithms (e.g., genetic algorithms) or local optimization algorithms (e.g., gradient descent).

[0117] A brief introduction to the execution process of the optimization algorithm is as follows.

[0118] The genetic algorithm is suitable for optimization problems in complex scenes, and its specific optimization process is as follows.

[0119] 1. Initial Population Generation: Randomly generate a set of possible solutions, with each solution consisting of the position and pose parameters of the virtual content.

[0120] 2. Fitness Function Calculation: Calculate the objective function EEE for each solution, using it as the fitness value.

[0121] 3. Selection, Crossover, and Mutation:

- [0122] Selection: Select the better solutions based on their fitness values.
- [0123] Crossover: Perform crossover operations on the selected solutions to generate new solutions.
- [0124] Mutation: Apply mutation operations to the new solutions to increase diversity.
- [0125] 4. Iteration: Repeat the steps of fitness function calculation, selection, crossover, and mutation until the maximum number of iterations is reached or the error converges.
- [0126] Gradient descent is a local optimization algorithm that iteratively updates parameters to minimize the objective function. The specific optimization process is as follows.
- [0127] 1. Parameter Initialization: Set the initial position and pose parameters.
- [0128] 2. Gradient Calculation: Calculate the gradient of the objective function E with respect to the position and pose parameters.
- [0129] 3. Parameter Update: Update the position and pose parameters based on the gradient and learning rate.
- [0130] 4. Iteration: Repeat the steps of gradient calculation and parameter update until the error converges or the maximum number of iterations is reached.
- [0131] The position and pose parameters output by the above optimization algorithms are used to adjust the position and pose of the virtual content, ensuring precise alignment between the virtual content and the real object. This step includes the following sub-steps.
- [0132] The virtual content is moved to the corresponding position based on the optimized position information. According to the optimized pose information, the virtual content is rotated so that the normal vector of the virtual content aligns with the normal vector of the real object. The specific adjustment process is as follows.
- [0133] As shown in Fig. 3, when there are two physical markers in the real-world scene, Step (S305) includes the following sub-steps.
- [0134] a1. The position of the real-world scene is calculated.
- [0135] Based on the known positions of the physical marker A and the physical marker C, the length of AC is calculated.

[0136] Using the known pose of A and the pose (quaternion) of C, $\vec{v}_{AB}$ and $\vec{v}_{BC}$ is obtained. Based on the formula: $\vec{v}_{AC} = \vec{v}_{AB} + \vec{v}_{BC}$, the $\vec{v}_{AC}$ is calculated.

[0137] Using $\frac{\vec{v}_{BC}}{\vec{v}_{AB}} = \tan(\theta)$, the angle between $\vec{v}_{AB}$ and $\vec{v}_{BC}$ is calculated.

[0138] Using $\sin(\theta) = \frac{BC}{AC}$ and the length of AC, the length of BC is calculated. The length of AB is further calculated.

[0139] Using the position of the physical marker A (PA), $\vec{v}_{AB}$ and length of AB, the position of B is calculated.

[0140] Using the position of the physical marker B, $\vec{v}_{BC}$, the length of BC and the position of the physical marker C, the position of C' (PC') is calculated.

[0141] The position of the real-world scene is represented by:

$$P_{\text{final}} = (PA + PC') / 2$$

[0142] a2. The rotation of the real-world scene is calculated as follows.

[0143] The normal line of the plane constructed by $\vec{v}_{AB}$ and $\vec{v}_{AC}$ is taken as the forward vector. The inverse of $\vec{v}_{BC}$ is taken as the upward vector. A quaternion (Q_final) is constructed by upward vector. and the forward vector. Through the above method, the position of the real-world scene and the pose of the real-world scene (Q_final) are calculated and assigned to the position and pose of the virtual content, achieving alignment between the virtual content and the real-world scene.

[0144] When there are three physical markers in the real-world scene, as shown in Fig. 4, Step (S305) includes the following sub-steps.

[0145] b1. The information of the physical markers is obtained.

[0146] Assume there are three positioning devices: A, B, and C, each providing its position data and pose data. The pose data of each physical marker is typically represented as QA、QB、QC, and the position data of each physical marker is typically represented as PA, PB and PC.

[0147] b2. The relative position between physical markers is calculated and the position of the real-world scene is determined.

[0148] According to the position of the positioning device C (PC) and the rotation of

the positioning device C (PQ), the position of C' (PC') is calculated. According to PC' and PA, the position of the real-world scene (P_final) is calculated as the following formula.

$$P_final = (PA + PC') / 2$$

[0149] b3. The rotation of the real-world scene is calculated.

[0150] The pose of the physical device (Q_final) can be determined by the forward vector, which is the normal line of the triangular plane constructed by the three points and, and the upward vector (the BC vector). Using these two vectors, a quaternion can be constructed. This allows for the calculation of the position and pose of the physical device, which is then assigned to the virtual content's position and pose, achieving alignment between the virtual content and the real-world scene.

[0151] In this embodiment of the application, the aspect ratio of the virtual content is determined based on the aspect ratio of the physical markers. The virtual content will be scaled proportionally according to the size of the real-world scene.

[0152] Embodiment 3

[0153] A system for displaying MR scenes provided in this embodiment, includes a point cloud acquisition module, a marker identification module and a registration and fusion module;

[0154] The point cloud acquisition module is configured to obtain a point cloud data of a real-world scene, wherein a plurality of physical markers are provided within the real-world scene, the plurality of physical markers has a three-dimensional structure with an uneven surface;

[0155] The marker identification module is configured to identify each of the plurality of physical markers through a classification algorithm; and to obtain a pose of each of the plurality of physical markers and a position of each of the plurality of physical markers, as well as a relative relationship between the plurality of physical markers, through point cloud registration; and

[0156] The registration and fusion module is configured to adjust a size of a virtual content, a pose of the virtual content and a position of the virtual content based on the

pose of each of the plurality of physical markers and the position of each of the plurality of physical markers as well as the relative relationship between the plurality of physical markers, and to align a virtual marker within the virtual content with the plurality of physical markers through point cloud registration, so as to achieve an alignment of the virtual content on a surface of a physical object.

[0157] It is understood that the system for displaying MR scene provided in this embodiment of the application corresponds to the aforementioned method for displaying MR scene. The explanations, examples, and beneficial effects related to this content can be referenced in the corresponding sections of the MR scene display method and will not be repeated here.

[0158] Embodiment 4

[0159] An electronic device, includes: a processor, a memory, and a program. The program is stored in the memory and configured to be executed by the processor, the program includes the method for displaying MR scenes as described as above.

[0160] In summary, compared to the prior art, the following beneficial effects of the present application are as follows.

[0161] 1. The embodiment adopts physical markers with the unique structure with uneven surfaces, providing rich geometric features such as corner points, edges, and curvature changes. These features are more easily identified and matched by point cloud data. By leveraging these prominent geometric features for feature extraction and characterization, the initial alignment accuracy of point cloud registration is improved. This enables the automatic identification and feature localization of real-world scenes, allowing virtual content to be accurately aligned and fit on their surfaces.

[0162] 2. The embodiment of the application increases the surface characteristics of physical markers by applying textured images to them. This further facilitates the identification and classification of physical markers in subsequent processes, improving classification accuracy.

[0163] 3. The embodiment of the application allows for the precise alignment of virtual content on the surface of real-world objects using just two physical markers. This not only simplifies installation but also reduces the complexity of point cloud

registration and fusion.

[0164] 4. The embodiment of the application introduces an adaptive real-time feedback mechanism that dynamically adjusts feedback frequency and optimization strategies based on changes in sensor data, ensuring continuous alignment of virtual content with the real-world scene.

[0165] 5. The embodiment of the application achieves a significant improvement in the alignment of virtual content with the real-world scene by selecting appropriate optimization algorithms and dynamically adjusting optimization parameters. In scenarios with large errors and complex environments, global optimization algorithms (such as genetic algorithms) are employed to find the optimal solution. In scenarios with small errors and simple environments, local optimization algorithms (such as gradient descent) are used for rapid convergence, increasing efficiency. Dynamically adjusting the optimization algorithm's parameters further enhances optimization effectiveness and the method's adaptability.

[0166] It should be noted that, relational terms such as "first" and "second" herein are used solely to distinguish one entity or operation from another and do not necessarily imply any actual relationship or order between these entities or operations. Moreover, the terms "including" or any other variations thereof are intended to cover non-exclusive inclusion, such that a process, method, article, or apparatus that includes a list of elements is not limited to those elements, but may also include other elements not explicitly listed, or elements that are inherent to such process, method, article, or apparatus. Without further limitation, an element defined by the phrase "including a ..." does not exclude the presence of additional identical elements in the process, method, article, or apparatus that includes the element.

[0167] Described above are merely preferred embodiments of this application, and are not intended to limit this application. Despite the detailed description of the present application with reference to the foregoing embodiments, it should be understood by those skilled in the art that any modifications, equivalent substitutions, and improvements made without departing from the spirit of this application shall fall in the scope of this application defined by the appended claims.

What is claimed is:

1. A method for displaying MR (mixed reality) scenes, comprising:

(S1) obtaining a point cloud data of a real-world scene, wherein a plurality of physical markers are provided within the real-world scene and have a three-dimensional structure with an uneven surface;

(S2) identifying each of the plurality of physical markers through a classification algorithm; and obtaining a pose of each of the plurality of physical markers and a position of each of the plurality of physical markers, as well as a relative relationship between the plurality of physical markers, through point cloud registration; and

(S3) adjusting a size of a virtual content, a pose of the virtual content and a position of the virtual content based on the pose of each of the plurality of physical markers and the position of each of the plurality of physical markers as well as the relative relationship between the plurality of physical markers, and aligning a virtual marker within the virtual content with the plurality of physical markers through point cloud registration, so as to achieve an alignment of the virtual content on a surface of a physical object.

2. The method for displaying MR scenes of claim 1, wherein the real-world scene has at least two physical markers, and a positional relationship between at least two physical markers among the plurality of physical markers satisfies the following condition:

a mounting surface of a first physical marker is on a first plane, and a mounting surface of a second physical marker is on a second plane; the first plane is perpendicular to the second plane;

the first plane intersects the second plane along a line; and

the closest point on the line to a center of the mounting surface of the first physical marker is the same as the closest point on the line to a center of the mounting surface of the second physical marker.

3. The method for displaying MR scenes of claim 1, wherein the plurality of physical markers are formed by stacking a plurality of cubes; so as to form the three-dimensional structure with the uneven surface.

4. The method for displaying MR scenes of claim 1, wherein the uneven surface comprises an image with a texture.

5. The method for displaying MR scenes of claim 1, wherein the step of identifying each of the plurality of physical markers by the classification algorithm comprises:

extracting a geometric feature and a surface feature from the point cloud data;

combining the geometric feature and the surface feature into feature vectors to characterize each of the plurality of physical markers; and

classifying the feature vectors by a machine learning algorithm to distinguish the plurality of physical markers.

6. The method for displaying MR scenes of claim 1, wherein the step of adjusting the size of the virtual content, the pose of the virtual content and the position of the virtual content based on the pose of each of the plurality of physical markers and the position of each of the plurality of physical markers as well as the relative relationship between the plurality of physical markers comprises:

calculating an overall change rate between a current frame and a previous frame of the real-world scene;

dynamically adjusting a feedback frequency based on the overall change rate, wherein the feedback frequency refers to a rate at which an alignment between the virtual content and the real-world scene is updated and adjusted according to the overall change rate;

calculating an alignment error between the virtual content and the real-world scene;

determining an optimization algorithm based on the alignment error and a scene complexity; and

adjusting the size of the virtual content, the position of the virtual content and the

pose of the virtual content according to the optimization algorithm.

7. The method for displaying MR scenes of claim 6, wherein the step of calculating the overall change rate between the current frame and the previous frame of the real-world scene comprises:

calculating a rate of change in the position of a physical marker among the plurality of the physical markers in the real-world scene between the current frame and the previous frame to obtain a position change rate, and calculating a rate of change in the pose of the physical marker in the real-world scene between the current frame and the previous frame to obtain a pose change rate; and

obtaining the overall change rate based on the position change rate and the pose change rate through the following formula:

$$R_c = \alpha \cdot R_p + \beta \cdot R_s;$$

wherein R_c represents the overall change rate; R_p represents the position change rate; R_s represents the pose change rate; and α and β are weighting coefficients.

8. The method for displaying MR scenes of claim 6, wherein the step of dynamically adjusting the feedback frequency based on the overall change rate comprises:

setting an initial feedback frequency, and dynamically adjusting the feedback frequency based on the overall change rate through the following formula:

$$F = F_b + (1 + k \times R_c)$$

wherein, F_b represents the initial feedback frequency, F represents the feedback frequency, k is an adjustment coefficient, and R_c represents the overall change rate.

9. The method for displaying MR scenes of claim 6, wherein the step of determining the optimization algorithm based on the alignment error and the scene complexity comprises:

setting a threshold for a positional error within the alignment error, a threshold for

an angular error within the alignment error and a threshold for the scene complexity;
and

determining whether the positional error is less than the threshold for the positional error, the angular error is less than the threshold for the angular error, and the scene complexity is less than the threshold for the scene complexity, if the above conditions are all met, determining a local optimization algorithm, otherwise, determining a global optimization algorithm.

10. A system for displaying MR scenes, comprising:

a point cloud acquisition module;

a marker identification module; and

a registration and fusion module;

wherein the point cloud acquisition module is configured to obtain a point cloud data of a real-world scene, wherein a plurality of physical markers are provided within the real-world scene, the plurality of physical markers has a three-dimensional structure with an uneven surface;

the marker identification module is configured to identify each of the plurality of physical markers through a classification algorithm; and to obtain a pose of each of the plurality of physical markers and a position of each of the plurality of physical markers, as well as a relative relationship between the plurality of physical markers, through point cloud registration; and

the registration and fusion module is configured to adjust a size of a virtual content, a pose of the virtual content and a position of the virtual content based on the pose of each of the plurality of physical markers and the position of each of the plurality of physical markers as well as the relative relationship between the plurality of physical markers, and to align a virtual marker within the virtual content with the plurality of physical markers through point cloud registration, so as to achieve an alignment of the virtual content on a surface of a physical object.

11. An electronic device, comprising:

a processor;

a memory; and

a program;

wherein the program is stored in the memory and configured to be executed by the processor, the program comprises the method for displaying MR scenes of claim 1.